

ITS
Institut
Teknologi
Sepuluh Nopember

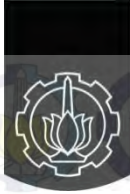
TUGAS AKHIR - TE141599

***Trajectory Tracking Gerak Cruise pada Quadcopter
Menggunakan Metode Linear Quadratic Tracking***

Mohammad Faizal Shultoni
NRP 2212 100 190

Dosen Pembimbing
Ir. Rusdhianto Effendie A.K., MT.
Eka Iskandar, ST. MT.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE141599

**CRUISE MOVEMENT TRAJECTORY TRACKING ON
QUADCOPTER USING LINEAR QUADRATIC TRACKING
METHOD**

Mohammad Faizal Shultoni
NRP 2212 100 190

Supervisor
Ir. Rusdhianto Effendie A.K., MT.
Eka Iskandar, ST., MT.

Electrical Engineering Department
Faculty of Industrial Engineering
Institut Teknologi Sepuluh Nopember
Surabaya 2016

**TRAJECTORY TRACKING GERAK CRUISE PADA
QUADCOPTER MENGGUNAKAN METODE LINEAR
QUADRATIC TRACKING**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik**

**Pada
Bidang Studi Teknik Sistem Pengaturan
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya**

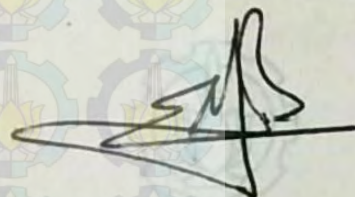
Menyetujui:

Dosen Pembimbing I

Dosen Pembimbing II



Ir. Rusdhianto Effendie A.K., MT.
NIP. 195704241985021001



Eka Iskandar, ST., MT.
NIP. 198005282008121001



Trajectory Tracking Gerak Cruise pada Quadcopter Menggunakan Metode Linear Quadratic Tracking

Nama : Mohammad Faizal Shultoni
Pembimbing I : Ir. Rusdhianto Effendie A.K., MT.
Pembimbing II : Eka Iskandar, ST., MT.

ABSTRAK

Gerakan *cruise* pada *quadcopter* dapat dilakukan bila *quadcopter* memiliki kestabilan dalam melakukan gerak rotasi dan gerak translasi. Dalam simulasi diusahakan respon gerak rotasi lebih cepat daripada respon gerak translasi. Pada *quadcopter* ini, perubahan sudut *pitch* mempengaruhi posisi *quadcopter* pada sumbu X, sedangkan perubahan sudut *roll* mempengaruhi posisi *quadcopter* pada posisi Y. Selain itu *quadcopter* memiliki system yang *non-linear* dan memiliki kestabilan yang rentan terhadap gangguan. Pada penelitian Tugas Akhir ini dirancang pengendalian gerak *cruise quadcopter* menggunakan kontroler *Proportioinal Integral Derivative* (PID) untuk gerak rotasi dan *Linear Quadratic Tracking* (LQT) untuk pengendalian gerak translasi. Untuk mendapatkan parameter dari PID digunakan cara *procedural method* dan untuk LQT digunakan metode *tuning try and error*. Dari *tuning* tersebut didapatkan parameter LQT $Q=100$ dan $R=0,1$ serta mendapatkan respon yang dengan *Error RMS* pada sumbu X sebesar 3,05% dan *Error* pada sumbu Y sebesar 2,49% untuk lintasan lingkaran. Sedangkan, untuk lintasan segi empat didapatkan *error RMS* pada sumbu X dan sumbu Y sebesar 1,58%. Dengan hasil tersebut *quadcopter* mampu melakukan *tracking trajectory* berbentuk lingkaran dan segi empat dengan baik.

Kata Kunci: *Quadcopter, Cruise, Proportional Integral Derivative, Linear Quadratic Tracking*



Cruise Movement Trajectory Tracking on Quadcopter using Linear Quadratic Tracking Method

Name : Mohammad Faizal Shultoni
Supervisor I : Ir. Rusdhianto Effendie A.K., MT.
Supervisor II : Eka Iskandar, ST., MT.

ABSTRACT

Cruise movement in quadcopter can be done if the quadcopter has stability while doing rotation and translation. In the simulation rotation response is being cultivated become faster than translation response. This quadcopter, transformation of pitch angle influence the position of the quadcopter by X axis, while the transformation of roll angle influence the position of quadcopter by Y axis. Beside that, quadcopter has non-linear system and has susceptible stability. In this research designed cruise movement control on quadcopter using Proportional Integral Derivative controller for rotation and Linear Quadratic Tracking controller for translation. To get the parameter of PID and LQT be used try and error method. By that tuning, found the parameter of LQT, $Q=100$ and $R=0.1$, and then got the response with RMS Error on X axis in the amount of 3.05% and RMS Error on Y axis 2.49% for circle track. While, the rectangle track has RMS Error on X and Y axis as big as 1.58%. By that result quadcopter can do circle and rectangle shape of tracking trajectory.

Keywords: *Quadcopter, Cruise, Proportional Integral Derivative, Linear Quadratic Tracking*



DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN	iii
HALAMAN PENGESAHAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Permasalahan.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Metodologi.....	3
1.6 Sistematika Penulisan.....	4
1.7 Relevansi.....	5
BAB II TEORI PENUNJANG	7
2.1 <i>Quadcopter</i>	7
2.1.1 Konsep Dasar <i>Quadcopter</i>	8
2.1.2 Kinematika <i>Quadcopter</i>	11
2.1.3 Dinamika <i>Quadcopter</i>	17
2.1.4 Model Matematika.....	24
2.2 Kontroler.....	25
2.2.1 <i>Proportional Integral Derivative (PID)</i>	26
2.2.2 <i>Linear Quadratic Tracking (LQT)</i>	28
2.3 Linearisasi.....	29
2.4 Ardupilot Mega 2.6.....	30
2.5 <i>Transmitter dan Receiver Radio</i>	31
2.6 <i>Remote Control Turnigy</i>	32
2.7 <i>Electronic Speed Controller (ESC)</i>	33
2.8 Motor BLDC dan <i>Propeller</i>	34
2.9 <i>Ground Station</i>	34

BAB III PERANCANGAN SISTEM	35
3.1 Spesifikasi Sistem.....	35
3.2 Identifikasi Kebutuhan.....	35
3.3 Perancangan Mekanik.....	35
3.4 Perancangan Elektronik.....	36
3.5 Pemodelan Motor dan <i>Propeller</i>	37
3.5.1 Pengukuran Kecepatan Motor.....	37
3.5.2 Pengukuran Gaya Angkat Motor.....	38
3.6 Identifikasi dan Validasi Sistem.....	38
3.7 Model Matematika Hasil Identifikasi.....	43
3.8 Perancangan <i>Proportional Integral Derivative</i> (PID).....	43
3.8.1 Perancangan Kontroler PID Sudut <i>Roll</i>	44
3.8.2 Perancangan Kontroler PID Sudut <i>Pitch</i>	45
3.8.3 Perancangan Kontroler PID Sudut <i>Yaw</i>	46
3.9 Perancangan <i>Linear Quadratic Tracking</i> (LQT).....	46
3.9.1 Perancangan Kontroler LQT Pengendalian Sumbu X.....	46
3.9.2 Perancangan Kontroler LQT Pengendalian Sumbu Y.....	48
3.9.3 Perancangan Kontroler LQT Pengendalian Sumbu Z.....	50
BAB IV PENGUJIAN DAN ANALISIS	53
4.1 Simulasi Pengujian PID dengan Nilai K_p Berbeda.....	53
4.1.1 Simulasi Pengujian PID Pengendalian Sudut <i>Roll</i> dengan Nilai K_p Berbeda.....	53
4.1.2 Simulasi Pengujian PID Pengendalian Sudut <i>Pitch</i> dengan Nilai K_p Berbeda.....	54
4.2 Simulasi Pengujian PID untuk Gerak Rotasi <i>Quadcopter</i>	56
4.2.1 Simulasi Pengujian PID untuk Pengendalian Sudut <i>Roll</i>	56
4.2.2 Simulasi Pengujian PID untuk Pengendalian Sudut <i>Pitch</i>	58
4.3 Simulasi Pengujian LQT dengan Nilai Q Berbeda pada Gerak <i>Cruise Quadcopter</i>	59
4.4 Simulasi Pengujian Gerak <i>Cruise Quadcopter</i> pada Lintasan Berbentuk Lingkaran dan Segi Empat.....	63
4.5 Simulasi Pengujian Gerak <i>Cruise Quadcopter</i> pada Lintasan Berbentuk Lingkaran dan Segi Empat dengan <i>Disturbance</i>	67

4.6	Simulasi 3D Gerak <i>Cruise Quadcopter</i> pada Lintasan Berbentuk Lingkaran dan Segi Empat	71
BAB V PENUTUP		75
5.1	Kesimpulan	75
5.2	Saran	76
DAFTAR PUSTAKA		77
LAMPIRAN A		A1
A.1	Pengukuran Kecepatan Motor	A1
A.2	Pengukuran Gaya Angkat Motor	A2
A.3	Identifikasi Fisik <i>Quadcopter</i>	A3
LAMPIRAN B		B1
B.1	Simulink Keseluruhan Sistem	B1
B.2	Simulink Pengendalian Sudut <i>Roll</i>	B2
B.3	Simulink Pengendalian Sudut <i>Pitch</i>	B2
B.4	Simulink Pengendalian Sudut <i>Yaw</i>	B3
B.5	Simulink Pengendalian Posisi X	B3
B.6	Simulink Pengendalian Posisi Y	B3
B.7	Simulink Pengendalian Posisi Z	B4
B.8	Simulink <i>Cascade</i> LQT PID Sumbu X	B4
B.9	Simulink <i>Cascade</i> LQT PID Sumbu Y	B4
LAMPIRAN C		C1
C.1	Program PID LQT pada MATLAB	C1
RIWAYAT PENULIS		D1

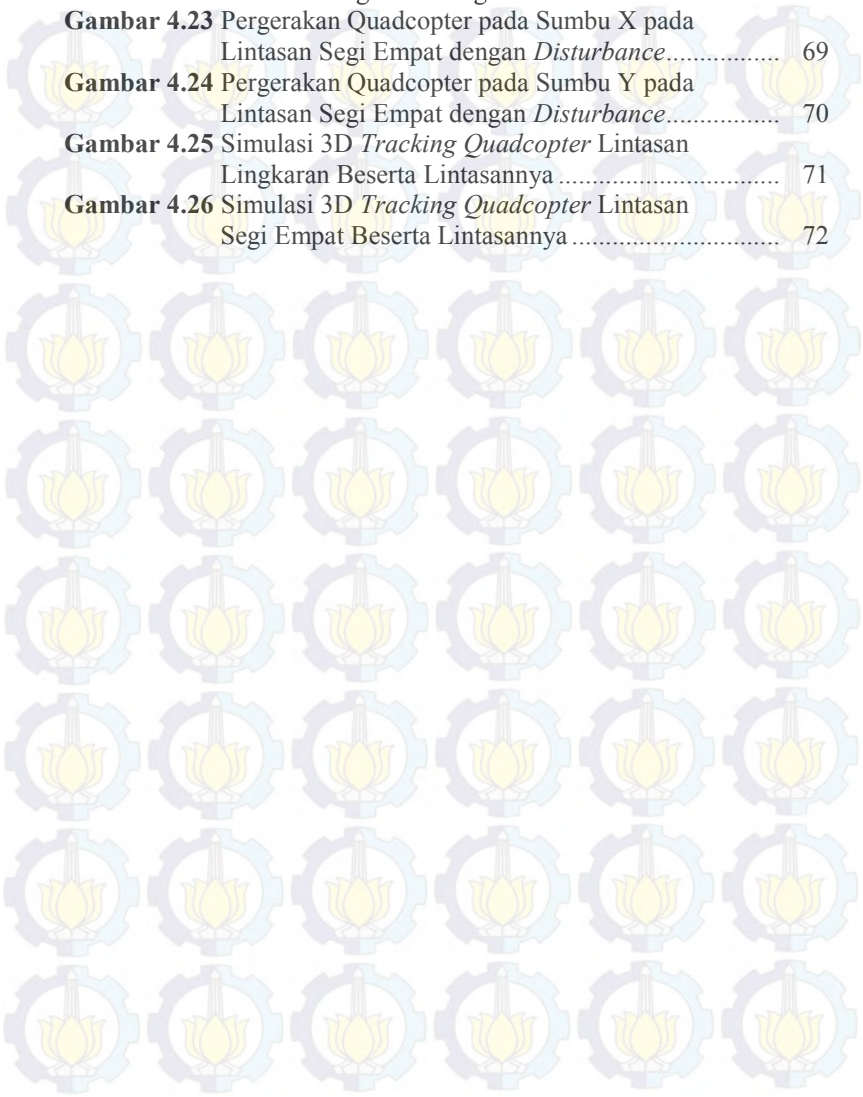


DAFTAR GAMBAR

Gambar 2.1 Konfigurasi <i>Quadcopter</i>	8
Gambar 2.2 Konfigurasi Arah Putaran Untuk Membangkitkan Gaya <i>Thrust</i>	9
Gambar 2.3 Konfigurasi Arah Putaran Untuk Membangkitkan Gaya <i>Roll</i>	9
Gambar 2.4 Konfigurasi Arah Putaran Untuk Membangkitkan Gaya <i>Pitch</i>	10
Gambar 2.5 Konfigurasi Arah Putaran Untuk Membangkitkan Gaya <i>Yaw</i>	10
Gambar 2.6 <i>Earth-frame</i> dan <i>Body-frame</i> dari <i>Quadcopter</i>	11
Gambar 2.7 Rotasi <i>Quadcopter</i> Sumbu Z	12
Gambar 2.8 Rotasi <i>Quadcopter</i> Sumbu Y	13
Gambar 2.9 Rotasi <i>Quadcopter</i> Sumbu X	14
Gambar 2.10 Blok Diagram <i>Closed Loop</i>	26
Gambar 2.11 Pengaturan <i>Loop</i> Tertutup Kontroler PID	26
Gambar 2.12 Blok Diagram LQT	29
Gambar 2.13 Ardupilot Mega 2.6	31
Gambar 2.14 <i>Transmitter</i> dan <i>Receiver</i>	31
Gambar 2.15 <i>Remote Control (Transmitter)</i> Turnigy 9XR	32
Gambar 2.16 ESC TBS Bulletproof 30 <i>Ampere</i>	33
Gambar 2.17 Motor <i>Quadcopter</i> dan <i>Propeller</i>	34
Gambar 2.18 <i>Telemetry</i> dan <i>Ground Station</i>	34
Gambar 3.1 Desain Mekanik <i>Quadcopter</i> yang Telah Dirancang ..	36
Gambar 3.2 Perancangan Sistem Pengendalian <i>Quadcopter</i>	37
Gambar 3.3 Blok Diagram Pengendalian Sudut <i>Roll</i>	44
Gambar 3.4 Blok Diagram Pengendalian Sudut <i>Pitch</i>	45
Gambar 3.5 Blok Diagram Pengendalian Gerak Translasi <i>Quadcopter</i> pada Posisi X	47
Gambar 3.6 Blok Diagram Pengendalian Gerak Translasi <i>Quadcopter</i> pada Posisi Y	48
Gambar 3.7 Blok Diagram Pengendalian Gerak Translasi <i>Quadcopter</i> pada Posisi Z	50
Gambar 4.1 Respon Variasi Nilai K_p pada Pengendalian PID Sudut <i>Roll</i>	54
Gambar 4.2 Respon Variasi Nilai K_p pada Pengendalian PID Sudut <i>Pitch</i>	55

Gambar 4.3	Respon Simulasi Sudut <i>Roll</i> dengan Referensi Berbeda	56
Gambar 4.4	Respon Simulasi Sudut <i>Roll</i> dengan Nilai Awal Berbeda	57
Gambar 4.5	Respon Simulasi Sudut <i>Pitch</i> dengan Referensi Berbeda	58
Gambar 4.6	Respon Simulasi Sudut <i>Pitch</i> dengan Nilai Awal Berbeda	58
Gambar 4.7	Respon Simulasi LQT Pergerakan <i>Quadcopter</i> pada Sumbu X dengan Menggunakan Bobot Q1	59
Gambar 4.8	Respon Simulasi LQT Pergerakan <i>Quadcopter</i> pada Sumbu Y dengan Menggunakan Bobot Q1	60
Gambar 4.9	Respon Simulasi LQT Pergerakan <i>Quadcopter</i> pada Sumbu X dengan Menggunakan Bobot Q2	61
Gambar 4.10	Respon Simulasi LQT Pergerakan <i>Quadcopter</i> pada Sumbu Y dengan Menggunakan Bobot Q2	61
Gambar 4.11	Respon Simulasi LQT Pergerakan <i>Quadcopter</i> pada Sumbu X dengan Menggunakan Bobot Q3	62
Gambar 4.12	Respon Simulasi LQT Pergerakan <i>Quadcopter</i> pada Sumbu Y dengan Menggunakan Bobot Q3	62
Gambar 4.13	Respon Simulasi Gerak <i>Cruise Quadcopter</i> pada Lintasan Berbentuk Lingkaran	64
Gambar 4.14	Pergerakan <i>Quadcopter</i> pada Sumbu X dengan Lintasan Lingkaran	64
Gambar 4.15	Pergerakan <i>Quadcopter</i> pada Sumbu Y dengan Lintasan Lingkaran	65
Gambar 4.16	Respon Simulasi Gerak <i>Cruise Quadcopter</i> pada Lintasan Berbentuk Segi Empat	65
Gambar 4.17	Pergerakan <i>Quadcopter</i> pada Sumbu X dengan Lintasan Segi Empat	66
Gambar 4.18	Pergerakan <i>Quadcopter</i> pada Sumbu Y dengan Lintasan Segi Empat	66
Gambar 4.19	Respon Simulasi Gerak <i>Cruise Quadcopter</i> pada Lintasan Berbentuk Lingkaran dengan <i>Disturbance</i> .	67
Gambar 4.20	Respon Simulasi Gerak <i>Cruise Quadcopter</i> pada Lintasan Berbentuk Segi Empat dengan <i>Disturbance</i>	67
Gambar 4.21	Pergerakan <i>Quadcopter</i> pada Sumbu X pada Lintasan Lingkaran dengan <i>Disturbance</i>	68

Gambar 4.22 Pergerakan Quadcopter pada Sumbu Y pada Lintasan Lingkaran dengan <i>Disturbance</i>	69
Gambar 4.23 Pergerakan Quadcopter pada Sumbu X pada Lintasan Segi Empat dengan <i>Disturbance</i>	69
Gambar 4.24 Pergerakan Quadcopter pada Sumbu Y pada Lintasan Segi Empat dengan <i>Disturbance</i>	70
Gambar 4.25 Simulasi 3D <i>Tracking Quadcopter</i> Lintasan Lingkaran Beserta Lintasannya	71
Gambar 4.26 Simulasi 3D <i>Tracking Quadcopter</i> Lintasan Segi Empat Beserta Lintasannya	72





--halaman ini sengaja dikosongkan--

DAFTAR TABEL

Tabel 3.1 Parameter <i>Quadcopter</i> Hasil Pengukuran	40
Tabel 3.2 Parameter <i>Quadcopter</i> Hasil Identifikasi Parametrik	42
Tabel 3.3 Parameter <i>Quadcopter</i> Hasil Identifikasi Fisik	42
Tabel 3.4 Validasi Hasil Identifikasi Parametrik dan Identifikasi Fisik	43
Tabel 3.5 Nilai K_p , K_i , dan K_d Kontroler PID Sudut <i>Roll</i>	44
Tabel 3.6 Nilai K_p , K_i , dan K_d Kontroler PID Sudut <i>Pitch</i>	45
Tabel 3.7 Nilai K_p , K_i , dan K_d Kontroler PID Sudut <i>Yaw</i>	46
Tabel 3.8 Nilai Q dan R Kontroler LQT Pengendalian Sumbu X , Y , dan Z	51
Tabel 4.1 Variasi Nilai K_p pada PID Pengendalian Sudut <i>Roll</i>	53
Tabel 4.2 Karakteristik Respon Variasi Nilai K_p pada PID Pengendalian Sudut <i>Roll</i>	54
Tabel 4.3 Variasi Nilai K_p pada PID Pengendalian Sudut <i>Pitch</i>	55
Tabel 4.4 Karakteristik Respon Variasi Nilai K_p pada PID Pengendalian Sudut <i>Pitch</i>	55
Tabel 4.5 Variasi Bobot Q Pada LQT Pengendalian Gerak <i>Cruise</i> ..	59
Tabel 4.6 Parameter Hasil <i>Tuning</i> dengan Bobot Q_1	60
Tabel 4.7 Parameter Hasil <i>Tuning</i> dengan Bobot Q_2	61
Tabel 4.8 Parameter Hasil <i>Tuning</i> dengan Bobot Q_3	63
Tabel 4.9 Parameter Pengujian <i>Quadcopter</i> Lintasan Lingkaran	63
Tabel 4.10 Parameter Pengujian <i>Quadcopter</i> Lintasan Segi Empat	67
Tabel 4.11 Parameter Pengujian <i>Quadcopter</i> Lintasan Lingkaran dengan <i>Disturbance</i>	70
Tabel 4.12 Parameter Pengujian <i>Quadcopter</i> Lintasan Segi Empat dengan <i>Disturbance</i>	70



BAB I

PENDAHULUAN

Tugas Akhir adalah suatu penelitian yang bersifat mandiri yang dilakukan sebagai persyaratan akademik untuk mendapatkan gelar sarjana teknik di Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Topik yang dibahas dalam Tugas Akhir ini ialah mengenai *trajectory tracking* gerak *cruise* pada *quadcopter*.

Pada Bab ini membahas mengenai hal-hal yang mendahului pelaksanaan Tugas Akhir. Hal tersebut meliputi latar belakang, permasalahan, batasan masalah, tujuan, metodologi, sistematika penulisan, dan relevansi.

1.1 Latar Belakang

Beberapa tahun terakhir, teknologi penerbangan berkembang dengan pesat. Teknologi tersebut dituntut untuk dapat membantu kinerja manusia. Salah satu contohnya ialah kendaraan udara tak berawak atau UAV (*Unmanned Aerial Vehicle*). UAV memiliki banyak kegunaan dalam berbagai bidang khususnya pada bidang militer dan sipil. UAV memiliki berbagai jenis, yaitu *fixed wing*, *axial wing*, *coaxial wing*, dan *Rotary wing*.

Salah satu jenis *Rotary wing* adalah *Quadcopter*. *Quadcopter* sering disebut *Quadcopter* karena memiliki 4 propeller dengan konfigurasi menyilang. Propeller bagian depan dan belakang akan berputar searah dengan jarum jam (*clockwise*). Bagian kiri dan kanan akan berputar berlawanan arah jarum jam (*counter clockwise*).

Pergerakan *quadcopter* biasanya dikendalikan oleh *user* menggunakan *remote control*. Bila ingin menuju ke suatu tempat harus dikendalikan lewat *remote control*. Penggunaan alat ini biasanya tidak terlalu akurat bila menuju ke suatu tempat. Oleh karena itu untuk membuat pergerakan *quadcopter* menjadi lebih akurat, maka digunakan *Trajectory Tracking*. Dengan menggunakan *trajectory tracking* diharapkan *quadcopter* mampu menuju lokasi tujuan dalam waktu yang diinginkan. *Tracking* yang dilakukan *quadcopter* adalah *tracking* untuk gerakan *cruise*. Gerakan ini menuntuk *quadcopter* untuk mampu melakukan maneuver untuk mencapai tempat tujuannya.

Untuk memastikan *quadcopter* mampu mengikuti lintasan, digunakan metode *Linear Quadratic Tracking*. Metode ini merupakan sistem pengaturan yang keluarannya diatur agar mengikuti (*tracking*) jalur

yang telah ditetapkan melalui masukan. *Tracking* dilakukan pada *quadcopter* agar mengikuti respon masukan yang telah ditentukan sehingga simpangan yang terjadi sekecil mungkin.

Penerapan penelitian ini dapat digunakan pada *quadcopter* untuk melakukan navigasi, liputan, pemetaan wilayah, serta kegiatan lain yang membutuhkan *quadcopter* melakukan banyak gerakan manuver untuk menyelesaikannya.

1.2 Permasalahan

Pada plant *quadcopter* ini memiliki permasalahan untuk kestabilan *quadcopter* saat melakukan *tracking* lintasan. Sementara untuk bergerak ke suatu titik dengan tepat dibutuhkan titik tujuan (*waypoint*) yang akan dijadikan referensi yang akan dilalui *quadcopter*. Dalam penelitian ini diinginkan *quadcopter* untuk mampu bergerak *cruise* dengan stabil dari satu titik awal menuju titik tujuan. *Quadcopter* diusahakan mengikuti lintasan atau *tracking* yang telah ditentukan, sebisa mungkin agar simpangan yang terjadi antara *quadcopter* dengan lintasan kecil, sehingga gerak *quadcopter* sesuai dengan lintasan yang ditentukan. Dalam hal ini berarti permasalahannya adalah optimalisasi posisi. *Heading* atau arah hadap dari *quadcopter* harus sesuai dengan yang ditentukan. Pada penelitian ini digunakan *Trajectory Tracking* untuk menentukan lintasan yang akan dilintasi *quadcopter* dari satu titik ke titik tujuan. Untuk memastikan *quadcopter* mengikuti lintasan yang ditentukan, digunakan metode *Linear Quadratic Tracking (LQT)*.

1.3 Batasan Masalah

Untuk membuat suatu penelitian diperlukan suatu batasan agar penelitian dapat lebih spesifik. Pada penelitian Tugas Akhir ini digunakan model UAV jenis *quadcopter* rakitan yang berada pada Laboratorium Teknik Pengaturan B-204, Jurusan Teknik Elektro ITS.

Frame yang digunakan ialah *frame talon turnigy* di mana dirancang dengan konfigurasi *plus (+)*. Berat dari *quadcopter* 1,26 kg dengan jari-jari 20,6 cm. Kontroler yang digunakan ialah Ardupilot Mega 2.6 dengan mikrokontroler ATmega 2560. Sensor *gyro* dan *accelerometer* yang digunakan ialah MPU 6050 yang terdapat dalam Ardupilot Mega 2.6. *Ground Station* yang digunakan ialah *Mission Planner*. Komunikasi dengan *ground station* menggunakan telemetry 433 Mhz.

Pemodelan *quadcopter* dilakukan dengan identifikasi parametrik untuk mendapatkan model matematika gerak rotasi *quadcopter*. Pengendalian dilakukan pada gerak *cruise* yaitu gerakan pada sumbu X, sumbu Y, dan sumbu Z. Variabel yang dikontrol berjumlah 6 yaitu posisi *quadcopter* pada sumbu X, posisi *quadcopter* pada sumbu Y, posisi *quadcopter* pada sumbu Z, sudut *pitch*, sudut *roll*, dan sudut *yaw* dari *quadcopter*.

Pada pengendalian posisi *quadcopter* digunakan metode *Linear Quadratic Tracking* (LQT) sedangkan pada pengendalian sudut *quadcopter* digunakan metode *Proportional-Integral-Derivative* (PID). Penentuan parameter K_p , K_i , dan K_d dari PID dilakukan dengan cara *procedural tuning* PID atau dengan kata lain sesuai dengan konseptual model. Untuk kontroler LQT, parameter Q dan R dilakukan dengan cara *tuning try dan error*. Perancangan kontroler dan simulasi dilakukan pada *software* MATLAB R2015a.

1.4 Tujuan

Dengan penelitian ini memiliki tujuan agar *quadcopter* bergerak menuju titik akhir tujuan melalui *waypoints* dengan gerak *cruise* dan dengan *heading quadcopter* yang ditentukan. *Quadcopter* dituntut untuk dapat mengikuti lintasan dengan seminimal mungkin mengalami simpangan. Oleh karena itu digunakan metode LQT karena, dengan persoalan optimasi posisi yang ada pada penelitian ini dianggap mampu diatasi oleh metode LQT. Dengan penelitian ini pula, penulis berharap penelitian ini dapat bermanfaat dalam penggunaan *quadcopter*, sehingga implementasi *quadcopter* di masa depan lebih luas.

1.5 Metodologi

Dalam penelitian Tugas Akhir ini diperlukan suatu tahapan yang merepresentasikan urutan yang harus dilaksanakan agar sesuai dengan tujuan penelitian. Tahapan tersebut ialah sebagai berikut:

a. Studi literatur

Pada tahap ini akan dilakukan kegiatan pengumpulan dan pengkajian hal-hal terkait teori, informasi maupun hasil eksperimen serupa yang dapat dijadikan referensi dalam proses penyusunan tugas akhir ini. Sumber yang dikumpulkan dan dikaji dapat diperoleh melalui berbagai sumber ilmiah seperti diktat, buku, hasil penelitian, maupun jurnal ilmiah yang telah dipublikasikan. Studi literatur difokuskan

mengenai konfigurasi dan cara kerja *quadcopter*, cara mapping terrain, *trajectory tracking* dan implementasi dari kontroler yang telah dibuat.

b. Identifikasi

Tahap yang kedua ialah mendapatkan model matematika dari *quadcopter*. Model matematika tersebut didapatkan dengan menggunakan identifikasi parametrik. Dengan didapatkannya model matematika dari *quadcopter*, perancangan kontroler dapat dilakukan.

c. Perancangan Kontroler

Pada tahap ini, telah didapatkan lintasan yang akan dilalui *quadcopter*, sehingga untuk memastikan *quadcopter* dapat melintasi lintasan dengan baik, maka dirancang kontroler. Pada *quadcopter* ini digunakan kontroler PID untuk rotasi, dan kontroler LQT untuk gerak translasi. Perancangan kontroler ini menggunakan software MATLAB R2015a.

d. Simulasi dan Evaluasi

Tahap yang keempat ialah kontroler yang telah dirancang menggunakan software MATLAB R2015a disimulasikan. Tujuan dari simulasi ialah untuk mengetahui performansi dari sistem yang telah diberi kontroler. Jika performansi yang didapatkan tidak sesuai dengan yang diinginkan maka dilakukan evaluasi. Evaluasi dilakukan dengan mengubah parameter dari kontroler yang telah dirancang.

e. Penulisan buku Tugas Akhir

Tahap yang terakhir ialah penulisan laporan/buku Tugas Akhir. Penulisan dilakukan secara intensif bila proses pengujian telah selesai.

1.6 Sistematika Penulisan

Tahap terakhir dari sebuah penelitian adalah penulisan laporan. Pada penulisan laporan/buku Tugas Akhir ini disusun berdasarkan 5 bab, di mana setiap bab berisi mengenai permasalahan dalam penelitian. Bab tersebut ialah sebagai berikut:

BAB I PENDAHULUAN

Berisi mengenai latar belakang, permasalahan, pembatasan masalah, tujuan, metodologi, sistematika penulisan, dan relevansi pembahasan tugas akhir ini.

BAB II TEORI PENUNJANG

Berisi mengenai konsep dasar dan teori yang mendasari perancangan tugas akhir ini, meliputi teori *quadcopter*, kinematika dan dinamika *quadcopter*, spesifikasi dari *quadcopter* yang digunakan, dan perancangan kontroler PID dan LQT.

BAB III PERANCANGAN SISTEM

Berisi mengenai spesifikasi sistem, identifikasi parameter, perancangan kontroler PID dan LQT dengan mengacu teori pada BAB II.

BAB IV PENGUJIAN DAN ANALISIS

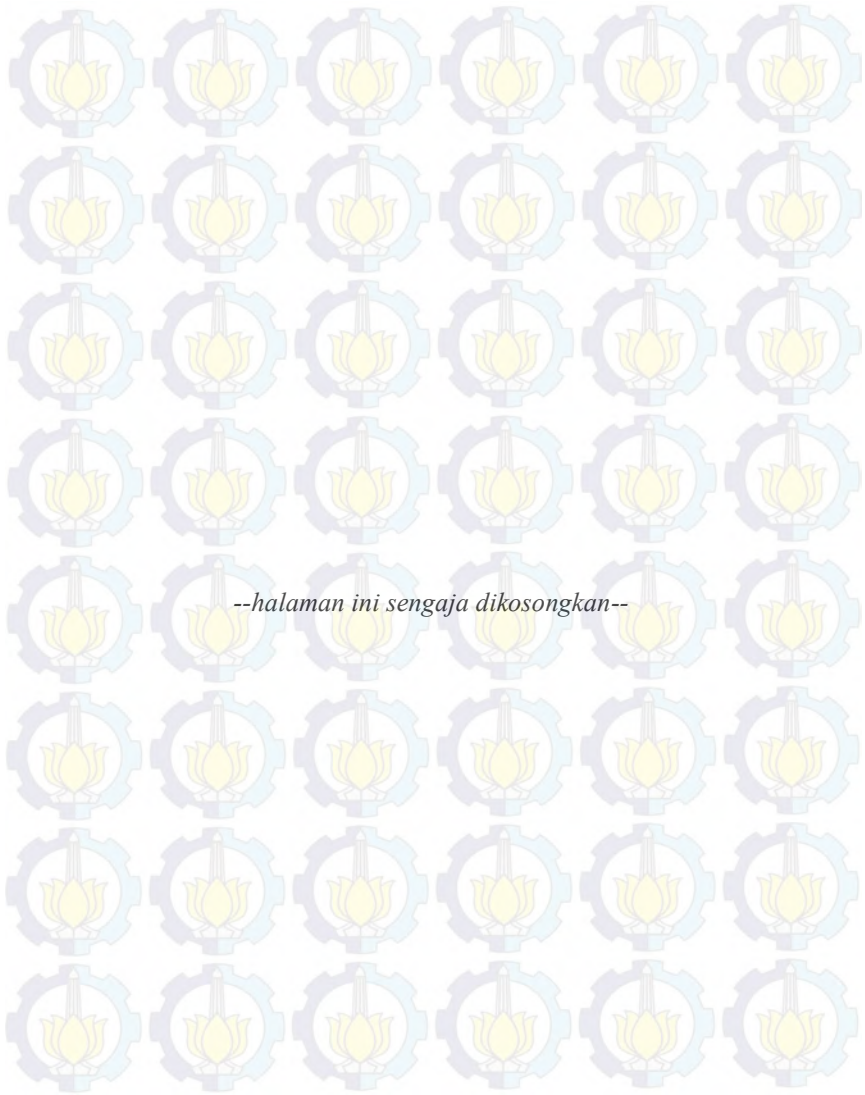
Berisi prosedur pelaksanaan pengujian dan analisis data mengenai pengendalian gerak *cruise* menggunakan kontroler PID dan LQT.

BAB V PENUTUP

Berisi mengenai kesimpulan dari penelitian Tugas Akhir dan saran untuk dapat digunakan untuk pengembangan tugas akhir ini untuk lebih lanjut.

1.7 Relevansi

Tugas akhir ini diharapkan dapat menjadi referensi untuk pengembangan UAV khususnya *quadcopter* untuk gerak *cruise*. Selain itu juga sebagai perbandingan metode kontrol yang lain untuk menentukan kontroler mana yang memiliki hasil paling baik.



BAB II

TEORI PENUNJANG

Suatu penelitian memerlukan teori-teori yang sudah ada sebelumnya untuk dikaji lebih dalam memperkuat argumen penulis. Teori tersebut digunakan untuk membantu penulis dan sebagai dasar dalam membuat suatu penelitian.

Pada bab ini terdapat beberapa teori dasar yang menjadi landasan untuk merumuskan dan menyelesaikan masalah yang akan dibahas pada penelitian ini. Pada bagian awal terdapat tinjauan pustaka yang menggambarkan landasan teori mengenai *quadcopter* secara umum yang akan digunakan pada tugas akhir ini. Pada bagian selanjutnya membahas mengenai teori-teori pendukung, meliputi dinamika dan kinematika *quadcopter*, kontroler PID dan LQT

2.1 *Quadcopter* [1] [2]

Unmanned Aerial Vehicle (UAV) menjadi perhatian utama dalam beberapa tahun terakhir. Hal ini dikarenakan perannya pada berbagai bidang seperti dalam hal keamanan, eksplorasi lingkungan, agrikultural, militer, pemetaan kota, dan lain sebagainya. Untuk menunjang aplikasi-aplikasi UAV yang sangat luas, diperlukan kemampuan *Vertical Take-Off* dan *Landing* (VOTL) dan gerak *hover* yang sesuai dengan kemampuan bergerak *quadcopter* yang cepat agar menjadi kendaraan yang ideal.

Setelah melalui banyak penelitian, tipe peralatan udara yang cocok untuk kondisi yang ada didalam ruangan dan diluar ruangan ialah *Rotorcraft* atau *Quadcopter Helicopter*. *Quadcopter* merupakan peralatan udara yang digerakkan oleh 4 baling-baling atau rotor. *Quadcopter* dapat dikontrol dengan mengatur kecepatan pada setiap rotornya sehingga mengubah besarnya gaya angkat dan torsi yang dihasilkan setiap rotor.

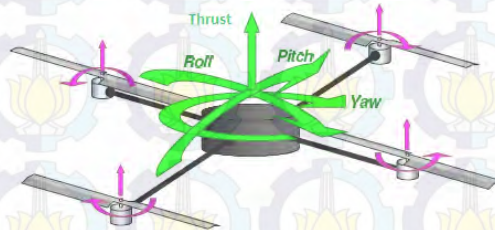
Quadcopter diklasifikasikan sebagai sebuah *Rotorcraft*, dan tidak termasuk dalam golongan *Fixed-wing*, sebab daya angkatnya diperoleh dari empat rotor. Daya angkat dari *Fixed-wing* diperoleh dari perbedaan aliran udara diantar bagian atas sayap dan bagian bawah sayap. Penggunaan empat rotor mengakibatkan setiap rotornya memiliki diameter yang lebih kecil jika dibandingkan dengan *Helicopter* dengan satu rotor. Penggunaan empat rotor ini akan mengurangi energi kinetik yang dihasilkan selama pesawat terbang, karena energi kinetik dipengaruhi oleh massa rotor dan kecepatannya. Bentuk rotor yang lebih kecil akan mengurangi resiko

adanya kerusakan yang disebabkan baling-baling mengenai obyek tertentu. Rotor dilindungi dari tabrakan yang keras dengan sebuah *frame* pelindung .

2.1.1 Konsep Dasar *Quadcopter* [2]

Quadcopter yang digunakan sebagai robot terbang kecil memiliki model mekanik yang terdiri dari empat *rotor* yang dipasang pada sumbu *plus* (+) simetris. Bentuk ini diharapkan tipis dan kaku, sehingga diperoleh friksi udara yang kecil dan komponen yang bergerak pada *quadcopter* hanyalah putaran *propeller*. Setiap *propeller* pada *quadcopter* diputar oleh satu motor elektrik, sehingga terdapat empat motor sebagai aktuator untuk menghasilkan gaya angkat dari *quadcopter*.

Dengan batasan menggunakan karakteristik motor dan *propeller* yang relatif sama, maka kondisi melayang (*hover*) akan diperoleh kecepatan motor yang sama disetiap *propeller*. Konfigurasi *propeller* terdiri dari dua macam, yaitu dua *propeller* yang bergerak searah jarum jam dan dua *propeller* yang bergerak berlawanan arah jarum jam di mana setiap satu sumbu, *propeller* berputar dengan arah yang sama yang terdapat pada Gambar 2.1.

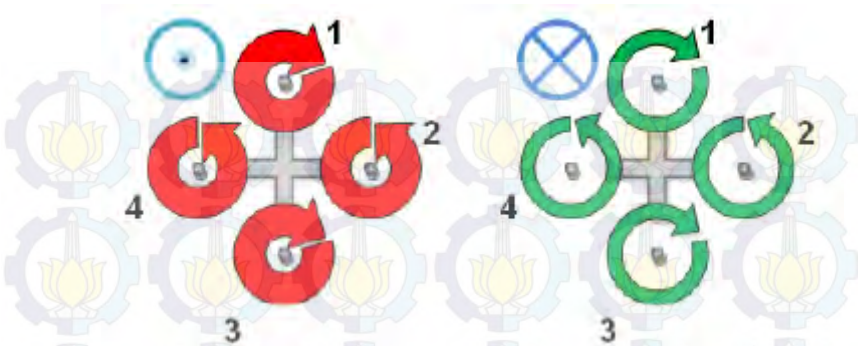


Gambar 2.1 Konfigurasi *Quadcopter*

Dengan melakukan pengaturan kecepatan putaran *propeller* akan dihasilkan beberapa komando *input*, diantaranya ialah:

a. Gaya *Thrust* (U_1)

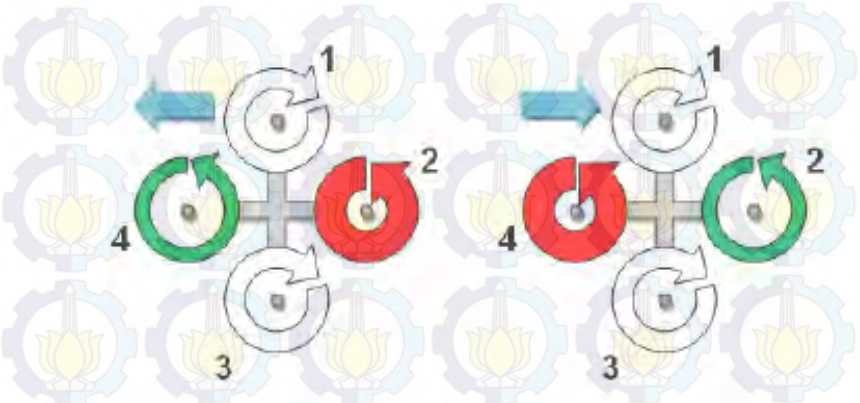
Dengan mempercepat (warna merah) atau memperlambat (warna hijau) kecepatan motor dari propeler secara bersamaan akan menghasilkan percepatan vertikal yang ditunjukkan pada Gambar 2.2.



Gambar 2.2 Konfigurasi Arah Putaran Untuk Membangkitkan Gaya *Thrust*

b. Torsi *Roll* (U_2)

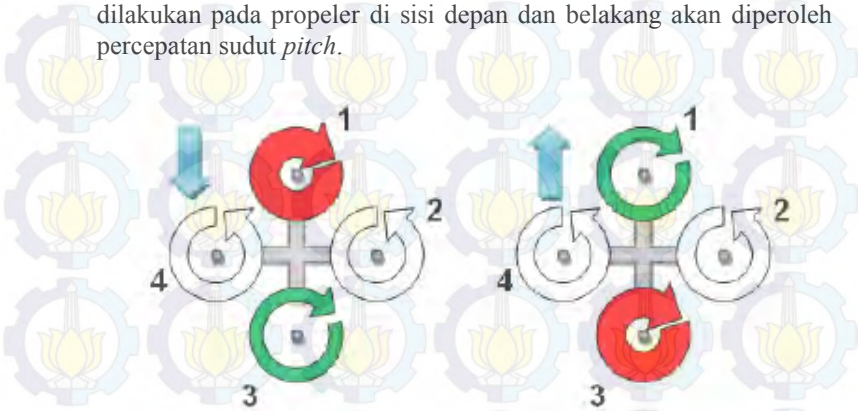
Dengan mempercepat atau memperlambat propeler pada sisi kiri dan secara bersamaan memperlambat atau mempercepat propeler di sisi kanan akan diperoleh rotasi dengan kecepatan sudut yang ditunjukkan pada Gambar 2.3.



Gambar 2.3 Konfigurasi Arah Putaran Untuk Membangkitkan Gaya *Roll*

c. Torsi *Pitch* (U_3)

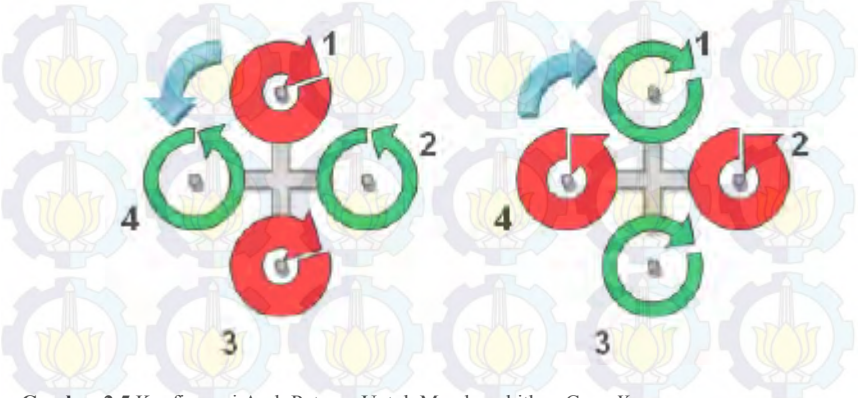
Pada Gambar 2.4 mekanisme yang sama seperti gerak *roll*, namun dilakukan pada propeler di sisi depan dan belakang akan diperoleh percepatan sudut *pitch*.



Gambar 2.4 Konfigurasi Arah Putaran Untuk Membangkitkan Gaya *Pitch*

d. Torsi *Yaw* (U_4)

Dengan mempercepat atau memperlambat propeler yang berputar berlawanan arah jarum jam, dan memperlambat atau mempercepat propeler yang berputar searah jarum jam akan diperoleh gerak *yaw* dengan percepatan sudut. Gerakan ini berputar dengan acuan pada sumbu Z. Gerakan *yaw* ditunjukkan pada Gambar 2.5.



Gambar 2.5 Konfigurasi Arah Putaran Untuk Membangkitkan Gaya *Yaw*

2.1.2 Kinematika *Quadcopter* [2]

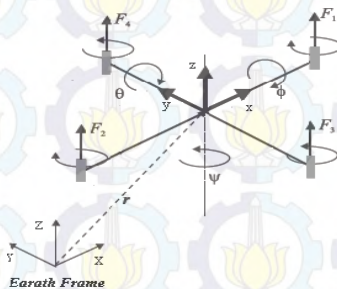
Sebelum melakukan analisis kinematika, dilakukan beberapa penetapan-penetapan yang akan digunakan. Pertama, analisis yang digunakan untuk kinematika memakai bingkai diagram cartesius tiga dimensi (X, Y, Z). Bingkai diagram cartesius dibagi menjadi dua, yaitu bingkai bumi (*earth frame*) yang kaku atau tidak bergerak, serta bingkai *body quadcopter* yang bergerak rotasi dan translasi. Ditetapkan sumbu X-bumi berada pada garis utara-selatan, sumbu Y-bumi berada pada garis timur-barat, sumbu Z-bumi berada pada garis vertikal menuju pusat bumi, sumbu X-*quadcopter* berada pada garis depan-belakang *quadcopter*, sumbu Y-*quadcopter* berada pada garis kiri-kanan *quadcopter*, dan sumbu Z-*quadcopter* berada pada garis atas-bawah *quadcopter*.

Bila didefinisikan sumbu X^+ menuju utara dan depan *quadcopter*, sumbu Z^+ menjauhi pusat bumi dan menuju atas *quadcopter* maka berlandaskan aturan kaidah tangan kanan *cross vector* dapat ditentukan bahwa sumbu Y^+ menuju barat atau kiri *quadcopter* seperti yang ditampilkan pada Gambar 2.6.

Posisi linear *quadcopter* (Γ^E) ditentukan dari koordinat vektor antara origin *B-frame* serta origin dari *E-frame* dengan memperhatikan *E-frame*. Posisi angular *quadcopter* (Θ^E) ditentukan dari orientasi *B-frame* terhadap *E-frame*. Posisi linear dan posisi angular terdapat pada Persamaan (2.1)-(2.2).

$$\Gamma^E = [X \ Y \ Z]^T \quad (2.1)$$

$$\Theta^E = [\phi \ \theta \ \Psi]^T \quad (2.2)$$

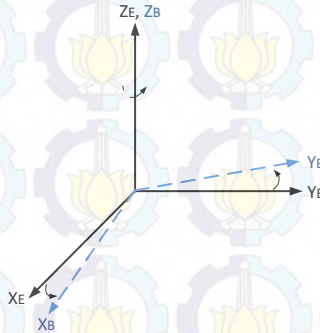


Gambar 2.6 Earth-frame dan Body-frame dari *Quadcopter*

Matriks rotasi dari *quadcopter* diperoleh dengan mengkalikan matriks transformasi untuk rotasi di tiap sumbu.

a. Rotasi Sumbu Z

Quadcopter berotasi pada sumbu Z dan menghasilkan sudut *yaw* yang dilambangkan dengan R (ψ , z). Rotasi sudut *yaw* terdapat pada Gambar 2.7



Gambar 2.7 Rotasi *Quadcopter* Sumbu Z

$$\begin{cases} X_E = X_B \cos \psi - Y_B \sin \psi \\ Y_E = X_B \sin \psi + Y_B \cos \psi \\ Z_E = Z_B \end{cases} \quad (2.3)$$

Persamaan (2.3) dapat dibuat dalam bentuk matriks pada Persamaan (2.4).

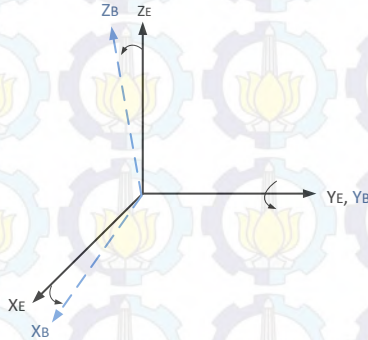
$$\begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} \quad (2.4)$$

Sehingga didapatkan matriks rotasi pada sumbu X ditunjukkan pada Persamaan (2.5).

$$R(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

b. Rotasi Sumbu Y

Quadcopter berotasi pada sumbu Y dan menghasilkan sudut *pitch* yang dilambangkan dengan $R(\theta, y)$. Rotasi sudut *pitch* terdapat pada Gambar 2.8



Gambar 2. 8 Rotasi *Quadcopter* Sumbu Y

$$\begin{cases} X_E = X_B \cos \theta + Z_B \sin \theta \\ Y_E = Y_B \\ Z_E = -X_B \sin \theta + Z_B \cos \theta \end{cases} \quad (2.6)$$

Persamaan (2.6) dapat dibuat dalam bentuk matriks menjadi Persamaan (2.7).

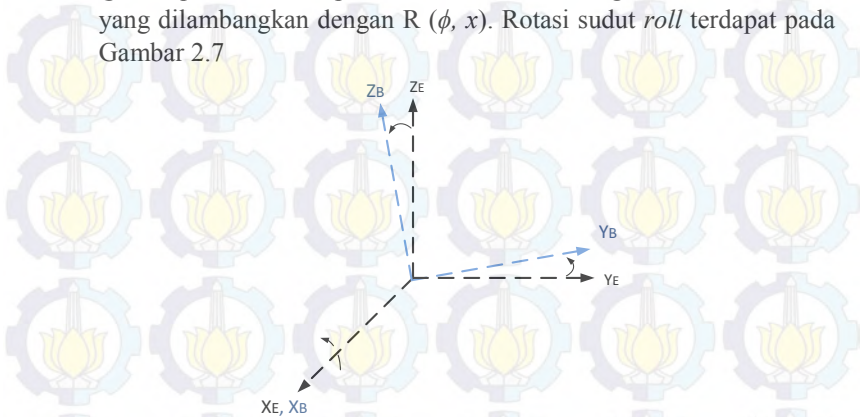
$$\begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} \quad (2.7)$$

Adapun matriks rotasi pada sumbu Y ditunjukkan pada Persamaan (2.8).

$$R(\theta, y) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \quad (2.8)$$

c. Rotasi Sumbu X

Quadcopter berotasi pada sumbu X dan menghasilkan sudut *roll* yang dilambangkan dengan $R(\phi, x)$. Rotasi sudut *roll* terdapat pada Gambar 2.7



Gambar 2. 9 Rotasi *Quadcopter* Sumbu X

$$\begin{cases} X_E = X_B \\ Y_E = Y_B \cos \phi - Z_B \sin \phi \\ Z_E = Y_B \sin \phi + Z_B \cos \phi \end{cases} \quad (2.9)$$

Persamaan (2.9) dapat dibuat dalam bentuk matriks menjadi Persamaan (2.10).

$$\begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} \quad (2.10)$$

Adapun matriks rotasi pada sumbu Z ditunjukkan pada Persamaan (2.11).

$$R(\phi, x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (2.11)$$

Sehingga persamaan rotasi *quadcopter* terdapat pada Persamaan (2.12)-(2.13).

$$R_{\Theta} = R(\phi) R(\theta) R(\psi) \quad (2.12)$$

$$R_{\Theta} = \begin{bmatrix} c_{\psi}c_{\theta} & -s_{\psi}c_{\phi} + c_{\psi}s_{\theta}s_{\phi} & s_{\psi}s_{\phi} + c_{\psi}s_{\theta}s_{\phi} \\ s_{\psi}c_{\theta} & c_{\psi}c_{\phi} + s_{\psi}s_{\theta}c_{\phi} & -c_{\psi}s_{\phi} + s_{\psi}s_{\theta}c_{\phi} \\ -s_{\theta} & c_{\theta}s_{\phi} & c_{\theta}c_{\phi} \end{bmatrix} \quad (2.13)$$

Sedangkan untuk kecepatan dalam *quadcopter* diekspresikan terhadap *body frame* (*B-frame*). Kecepatan *quadcopter* terdiri dari kecepatan linear V^B dan kecepatan angular ω^B . Komposisi vektornya disajikan dalam Persamaan (2.14)-(2.15).

$$\omega^B = [p \quad q \quad r]^T \quad (2.15)$$

Diperlukan kombinasi nilai linear dan angular untuk memberikan representasi yang lengkap dalam *space*. ξ merupakan komposisi dari vektor posisi linear Γ^E (m) dan vektor posisi sudut Θ^E (rad) terhadap *earth frame* (*E-frame*) seperti terlihat pada Persamaan (2.16).

$$\xi = [\Gamma^E \quad \Theta^E]^T = [X \quad Y \quad Z \quad \phi \quad \theta \quad \psi]^T \quad (2.16)$$

V merupakan generalisasi dari vektor kecepatan linear *quadcopter* V^B (m s^{-1}) dan kecepatan angular *quadcopter* ω^B (rad s^{-1}) pada *body frame* (*B-frame*) yang terdapat pada Persamaan (2.17)

$$V = [V^B \quad \omega^B]^T = [u \quad v \quad w \quad p \quad q \quad r]^T \quad (2.17)$$

Hubungan antara kecepatan linear pada *body frame* (*B-frame*) dan salah satu factor pada *earth frame* (*E-frame*) V^E (atau Γ^E) [m/s] dapat dilihat pada Persamaan (2.18).

$$V^E = \Gamma^E = R_{\Theta} V^B \quad (2.18)$$

Dimana R_{Θ} adalah matrik rotasi dari *body frame* (*B-frame*) ke *earth frame* (*E-frame*). Seperti pada kecepatan linear, hal tersebut juga berlaku untuk menghubungkan kecepatan angular pada *E-frame* (atau kecepatan Euler) $\dot{\Theta}^E$ [rad/s] ke *B-frame* ω^B atau sebaliknya. Hubungan tersebut terdapat pada Persamaan (2.19).

$$\omega^B = T_{\Theta}^{-1} \dot{\Theta}^E \quad (2.19)$$

$$\dot{\Theta}^E = T_{\Theta} \omega^B \quad (2.20)$$

Dimana T_{Θ} adalah matriks transformasi. Matriks transformasi T_{Θ} dapat ditetapkan dengan menggunakan kecepatan Euler dalam *B-frame*, dengan membalik pola perputaran sudut dari *roll*, *pitch* dan *yaw* seperti pada Persamaan (2.21).

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} + R(\phi)^{-1} \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + R(\phi)^{-1} R(\theta)^{-1} \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \quad (2.21)$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = T_{\Theta}^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (2.22)$$

Persamaan (2.13)-(2.19) maka diperoleh matriks transformasi dari *body frame* (*B-frame*) menuju *earth frame* (*E-frame*).

$$T_{\Theta}^{-1} = \begin{bmatrix} 1 & 0 & -s_{\theta} \\ 0 & c_{\phi} & c_{\theta} s_{\phi} \\ 0 & -s_{\phi} & c_{\phi} c_{\theta} \end{bmatrix} \quad (2.23)$$

$$T_{\Theta} = \begin{bmatrix} 1 & s_{\phi} t_{\theta} & c_{\phi} t_{\theta} \\ 0 & c_{\phi} & -s_{\phi} \\ 0 & s_{\phi} / c_{\theta} & c_{\phi} / c_{\theta} \end{bmatrix} \quad (2.24)$$

Persamaan yang sudah didapat, maka dibentuk suatu hubungan antara kecepatan terhadap *earth frame* (*E-frame*) dan *body frame* (*B-frame*).

$$\dot{\xi} = J_{\Theta} v \quad (2.25)$$

Dimana $\dot{\xi}$ adalah vektor kecepatan yang mengacu pada *earth frame* (*E-frame*), v adalah vektor kecepatan mengacu *body frame* (*B-frame*) dan J_{Θ} adalah matrik *jacobian*. Matrik *jacobian* terdiri dari 4 *sub-matrik* sebagaimana Persamaan (2.26).

$$J_{\Theta} = \begin{bmatrix} R_{\Theta} & 0_{3 \times 3} \\ 0_{3 \times 3} & T_{\Theta} \end{bmatrix} \quad (2.26)$$

2.1.3 Dinamika *Quadcopter* [2]

Berdasarkan aksioma pertama Euler dari hukum kedua Newton, turunan dari komponen linear dari gerakan suatu benda dapat dilihat pada Persamaan (2.25).

$$\left\{ \begin{array}{l} m \ddot{\Gamma}^E = F^E \\ m \overbrace{\dot{R}_{\Theta} V^B}^{\dot{R}_{\Theta} V^B} = R_{\Theta} F^B \\ m (R_{\Theta} \dot{V}^B + \dot{R}_{\Theta} V^B) = F^B \\ m R_{\Theta} (\dot{V}^B + \omega^B \times V^B) = R_{\Theta} F^B \\ m (\dot{V}^B + \omega^B \times V^B) = F^B \end{array} \right. \quad (2.25)$$

Dengan m (kg) adalah massa *quadcopter*, $\ddot{\Gamma}^E$ ($m s^{-2}$) adalah vektor percepatan linear *quadcopter* yang mengacu pada *E-frame*, F^E (N) adalah vektor gaya *quadcopter* terhadap *E-frame*, V^B ($m s^{-2}$) adalah percepatan linear *quadcopter* terhadap *B-frame*, dan \dot{R}_{Θ} adalah turunan pertama matriks rotasi. Dan simbol \times merupakan perkalian produk suatu vektor.

Berdasarkan aksioma kedua Euler dari hukum kedua Newton, dengan cara yang sama, turunan dari gerakan angular komponen dari suatu benda dapat dilihat pada Persamaan (2.26).

$$\left\{ \begin{array}{l} I \ddot{\Theta}^E = \tau^E \\ I \overbrace{\dot{T}_{\Theta} \omega^B}^{\dot{T}_{\Theta} \omega^B} = T_{\Theta} \tau^B \\ I \dot{\omega}^B + \omega^B \times (I \omega^B) = \tau^B \end{array} \right. \quad (2.26)$$

Pada Persamaan (2.26), I ($N m s^2$) adalah matriks inersia *quadcopter* (pada *B-frame*), $\ddot{\Theta}^E$ ($rad s^{-2}$) adalah vektor percepatan sudut *quadcopter*

terhadap E-frame, ω^B (rad s⁻²) adalah vektor percepatan sudut *quadcopter* terhadap B-frame, dan τ^E (N m) adalah torsi *quadcopter* terhadap E-frame.

Dari Persamaan (2.25) dan (2.26), dapat dibentuk persamaan benda kaku dengan 6 derajat kebebasan (DOF). Persamaan (2.27) menunjukkan formulasi matriks dari dinamika sistem.

$$\begin{bmatrix} m I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{bmatrix} \begin{bmatrix} \dot{V}^B \\ \omega^B \end{bmatrix} + \begin{bmatrix} \omega^B \times (mV^B) \\ \omega^B \times (I\omega^B) \end{bmatrix} = \begin{bmatrix} F^B \\ \tau^B \end{bmatrix} \quad (2.27)$$

Dimana notasi $I_{3 \times 3}$ adalah matriks identitas 3x3. Dan $0_{3 \times 3}$ adalah matriks nol 3x3. *quadcopter* memiliki empat buah motor yang menghasilkan gaya dorong seperti pada Gambar 2.6.

Vektor gaya yang terjadi pada *quadcopter* dapat ditentukan berdasarkan Persamaan (2.28).

$$\Lambda = [F^B \ \tau^B]^T = [F_x \ F_y \ F_z \ \tau_x \ \tau_y \ \tau_z]^T \quad (2.28)$$

Persamaan (2.28) dapat juga ditulis dalam bentuk formulasi matriks seperti pada Persamaan (2.29).

$$M_B \dot{v} + C_B(v)v = \Lambda \quad (2.29)$$

Dengan \dot{v} adalah vektor percepatan *quadcopter* terhadap B-frame. M_B adalah matriks inersia sistem dan C_B adalah matriks sentripetal Coriolis. Persamaan (2.30) menunjukkan matriks M_B .

$$M_B = \begin{bmatrix} m I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{bmatrix} = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{XX} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{YY} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{ZZ} \end{bmatrix} \quad (2.30)$$

Dapat dilihat bahwa matriks M_B adalah matriks diagonal, sedangkan matriks C_B ditunjukkan oleh Persamaan (2.31).

$$C_B = \begin{bmatrix} 0_{3 \times 3} & -m S(V_B) \\ 0_{3 \times 3} & -S(I \omega_B) \end{bmatrix}$$

$$\mathbf{C}_B = \begin{bmatrix} 0 & 0 & 0 & 0 & m w & -m v \\ 0 & 0 & 0 & -m w & 0 & m u \\ 0 & 0 & 0 & m v & -m u & 0 \\ 0 & 0 & 0 & 0 & I_{ZZ} r & -I_{YY} q \\ 0 & 0 & 0 & -I_{ZZ} r & 0 & I_{XX} p \\ 0 & 0 & 0 & I_{YY} q & -I_{XX} p & 0 \end{bmatrix} \quad (2.31)$$

Pada persamaan ini, diadopsi operator *skew-symmetric* $\mathbf{S}(\cdot)$. Untuk vektor 3 dimensi \mathbf{k} , matriks *skew-symmetric* dari \mathbf{k} ($\mathbf{S}(\mathbf{k})$) ditunjukkan pada Persamaan (2.32).

$$\mathbf{S}(\mathbf{k}) = -\mathbf{S}^T(\mathbf{k}) = \begin{bmatrix} 0 & -k_3 & k_1 \\ k_3 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{bmatrix}, \mathbf{k} = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} \quad (2.32)$$

Matriks $\mathbf{\Lambda}$ dapat dibagi menjadi tiga komponen menurut kontribusi sifat dasar *quadcopter*. Percepatan gravitasi hanya berlaku pada persamaan linear *quadcopter*. Persamaan matriks $\mathbf{G}_B(\xi)$ dituliskan pada Persamaan (2.33).

$$\begin{aligned} \mathbf{G}_B(\xi) &= \begin{bmatrix} \mathbf{F}_{GB} \\ \mathbf{0}_{3 \times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_\theta^{-1} \mathbf{F}_{GE} \\ \mathbf{0}_{3 \times 1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_\theta^T & \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \\ \mathbf{0}_{3 \times 1} \end{bmatrix} = \begin{bmatrix} mg \sin \theta \\ -mg \cos \theta \sin \phi \\ -mg \cos \theta \sin \phi \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \quad (2.33)$$

di mana \mathbf{F}_{GB} (N) adalah vektor gaya gravitasi terhadap B-frame dan \mathbf{F}_{GE} (N) terhadap E-frame. \mathbf{R}_θ adalah matriks ortogonal, sehingga matriks inversnya sama dengan matriks transposnya.

Kontribusi yang kedua mengenai efek giroskopis yang dihasilkan oleh perputaran *propeller*. Selama dua *propeller* yang berhadapan berputar searah jarum jam dan dua *propeller* lainnya berputar berlawanan arah jarum jam, terjadi ketidak-seimbangan saat jumlah aljabar dari kecepatan motor tidak sama dengan nol. Jika *roll* dan *pitch* juga tidak nol, *quadcopter* akan mengalami torsi efek giroskopis yang ditunjukkan pada Persamaan (2.34).

$$\begin{aligned}
\mathbf{O}_B(\mathbf{v})\boldsymbol{\Omega} &= \begin{bmatrix} 0_{3 \times 1} \\ -\sum_{k=1}^4 J_{TP} \left(\boldsymbol{\omega}^B \times \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ 1 \end{bmatrix} \right) (-1)^k \Omega_k \end{bmatrix} = \begin{bmatrix} 0_{3 \times 1} \\ J_{TP} \begin{bmatrix} -q \\ p \\ 0 \end{bmatrix} \boldsymbol{\Omega} \end{bmatrix} \\
&= J_{TP} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -q & -q & -q & -q \\ -p & p & -p & p \\ 0 & 0 & 0 & 0 \end{bmatrix} \boldsymbol{\Omega} \quad (2.34)
\end{aligned}$$

\mathbf{O}_B adalah matriks giroskopis dari *propeller* dan J_{TP} (N m^2) adalah momen inersia total di sekitar sumbu *propeller*. Efek giroskopis yang dihasilkan oleh putaran *propeller* hanya berhubungan dengan persamaan angular, dan tidak mempengaruhi persamaan linear. $\boldsymbol{\Omega}$ (rad s^{-1}) merupakan vektor kecepatan putar *propeller*, yang ditunjukkan oleh Persamaan (2.35).

$$\begin{aligned}
\boldsymbol{\Omega} &= -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4, \quad \boldsymbol{\Omega} = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} \quad (2.35)
\end{aligned}$$

di mana Ω_1 , Ω_2 , Ω_3 , dan Ω_4 (rad s^{-1}) adalah kecepatan *propeller* depan, kanan, belakang, dan kiri.

Kontribusi yang ketiga adalah perhitungan gaya dan torsi yang dihasilkan oleh pergerakan *input*. Pergerakan matriks \mathbf{E}_B dikali dengan $\boldsymbol{\Omega}^2$ untuk memperoleh vektor perpindahan $\mathbf{U}_B(\boldsymbol{\Omega})$. Efek aerodinamis (factor *thrust* b (N s^2) dan *drag* d (N m s^2)) berpengaruh pada gaya dan torsi yang dihasilkan. Persamaan vektor perpindahan pada dinamika *quadcopter* ditunjukkan pada Persamaan (2.36).

$$\begin{aligned}
\mathbf{U}_B(\boldsymbol{\Omega}) = \mathbf{E}_B \boldsymbol{\Omega}^2 &= \begin{bmatrix} 0 \\ 0 \\ U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ bl(-\Omega_2^2 + \Omega_4^2) \\ bl(-\Omega_1^2 + \Omega_3^2) \\ d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{bmatrix} \quad (2.36)
\end{aligned}$$

di mana l (m) adalah jarak antara pusat *quadcopter* dengan pusat *propeller*. U_1, U_2, U_3 , dan U_4 adalah komponen vektor gerak. Hubungan komponen vektor tersebut dengan *propeller* dapat diperoleh dari perhitungan aerodinamis.

Dari persamaan sebelumnya, dapat diketahui konstanta matriks E_B yang dikalikan dengan kuadrat dari kecepatan *propeller* Ω^2 sehingga menghasilkan $U_B(\Omega)$. Persamaan (2.37) menunjukkan matriks E_B .

$$E_B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ b & b & b & b \\ 0 & -bl & 0 & bl \\ -bl & 0 & bl & 0 \\ -d & d & -d & d \end{bmatrix} \quad (2.37)$$

Dari Persamaan (2.29), dinamika *quadcopter* dapat dibuat dengan mempertimbangkan ketiga kontribusi menurut Persamaan (2.38).

$$M_B \dot{v} + C_B(v)v = G_B(\xi) + O_B(v)\Omega + E_B\Omega^2 \quad (2.38)$$

Dengan mengatur ulang Persamaan (2.38) diatas, turunan dari vektor kecepatan terhadap B-frame dapat dirumuskan sebagai berikut:

$$\dot{v} = M_B^{-1} (C_B(v)v + G_B(\xi) + O_B(v)\Omega + E_B\Omega^2) \quad (2.39)$$

Persamaan (2.40) menunjukkan persamaan-persamaan sebelumnya bukan dalam bentuk matriks, melainkan dalam bentuk sistem Persamaan (2.40).

$$\begin{cases} \dot{u} = (vr - wq) + g s_\theta \\ \dot{v} = (wp - ur) - g c_\theta s_\phi \\ \dot{w} = (uq - vp) - g c_\theta s_\phi + \frac{U_1}{m} \\ \dot{p} = \frac{I_{YY} - I_{ZZ}}{I_{XX}} q r - \frac{J_{TP}}{I_{XX}} q \Omega + \frac{U_2}{I_{XX}} \\ \dot{q} = \frac{I_{ZZ} - I_{XX}}{I_{YY}} p r - \frac{J_{TP}}{I_{YY}} p \Omega + \frac{U_3}{I_{YY}} \\ \dot{r} = \frac{I_{XX} - I_{YY}}{I_{ZZ}} p q - \frac{U_4}{I_{ZZ}} \end{cases} \quad (2.40)$$

Persamaan *input* kecepatan *propeller* (balong-balong) dapat dilihat dalam Persamaan (2.41).

$$\begin{cases} U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ U_2 = bl(-\Omega_2^2 + \Omega_4^2) \\ U_3 = bl(-\Omega_1^2 + \Omega_3^2) \\ U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\ \Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \end{cases} \quad (2.41)$$

Persamaan sistem dinamik *quadcopter* (2.40) diperoleh dari *B-frame*. Dalam pemodelannya, dibutuhkan persamaan dinamik sistem yang mencakup sistem *hybrid* yang disusun oleh persamaan linear dari *E-frame* dan persamaan angular dari *B-frame*. Hal ini dilakukan untuk mempermudah pengaturan. Persamaan-persamaan berikut akan disajikan dalam kerangka “*hybrid*”, atau *H-frame*. Persamaan (2.42) menunjukkan vektor kecepatan *quadcopter* yang digeneralisasi pada *H-frame*.

$$\dot{\xi} = [\Gamma^E \quad \omega^B]^T = [\dot{X} \quad \dot{Y} \quad \dot{Z} \quad p \quad q \quad r]^T \quad (2.42)$$

Dinamika sistem pada *H-frame* dapat dituliskan dalam bentuk matriks menurut Persamaan (2.43).

$$\mathbf{M}_H \dot{\xi} + \mathbf{C}_H(\xi)\xi = \mathbf{G}_H + \mathbf{O}_H(\xi) \Omega + \mathbf{E}_H(\xi)\Omega^2 \quad (2.43)$$

Berikut ini akan ditentukan semua matriks dan vektor yang ditunjukkan oleh persamaan di atas. Matriks inersia sistem terhadap *H-frame* \mathbf{M}_H sama dengan matriks inersia sistem terhadap *B-frame*, dan ditentukan seperti pada Persamaan (2.44).

$$\mathbf{M}_H = \mathbf{M}_B = \begin{bmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{XX} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{YY} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{ZZ} \end{bmatrix} \quad (2.44)$$

Namun, matriks sentripetal Coriolis terhadap H-frame $C_H(\xi)$ tidak sama dengan matriks sentripetal Coriolis terhadap B-frame dan ditentukan seperti pada Persamaan (2.45).

$$C_H(\xi) = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & -S(I \omega_B) \end{bmatrix} \quad (2.45)$$

Vektor gravitasi terhadap H-frame G_H dituliskan pada Persamaan (2.46). Dapat dilihat bahwa gravitasi mempengaruhi ketiga persamaan linear, namun lebih berpengaruh terhadap ketinggian *quadcopter*.

$$G_H = \begin{bmatrix} F_G^E \\ 0_{3 \times 1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.46)$$

Efek giroskopis yang dihasilkan oleh putaran *propeller* tidak berubah karena hanya mempengaruhi persamaan angular yang mengacu kepada B-frame. Oleh karena itu, matriks giroskopis terhadap H-frame $O_H(\xi)$ dibuat sama dengan Persamaan (2.47).

$$\begin{aligned} O_H(\xi)\Omega &= O_B(v)\Omega = \begin{bmatrix} 0_{3 \times 1} \\ J_{TP} \begin{bmatrix} -q \\ p \\ 0 \end{bmatrix} \Omega \end{bmatrix} \\ &= J_{TP} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ q & -q & q & -q \\ -p & p & -p & p \\ 0 & 0 & 0 & 0 \end{bmatrix} \Omega \end{aligned} \quad (2.47)$$

Matriks perpindahan terhadap H-frame $E_H(\xi)$ berbeda dengan perpindahan terhadap B-frame karena *input* U_1 mempengaruhi semua persamaan linear dengan matriks rotasi R_θ . Perkalian produk antara matriks perpindahan dengan kuadrat kecepatan *propeller* ditunjukkan pada Persamaan (2.48).

$$\begin{aligned}
 E_H(\xi)\Omega^2 &= \begin{bmatrix} R_\theta & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & I_{3 \times 3} \end{bmatrix} E_B\Omega^2 \\
 &= \begin{bmatrix} (s_\psi s_\phi + c_\psi s_\theta c_\phi)U_1 \\ (-c_\psi s_\phi + s_\psi s_\theta c_\phi)U_1 \\ (c_\theta c_\phi)U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} \quad (2.48)
 \end{aligned}$$

Dengan menyusun kembali Persamaan (2.43), dapat ditemukan rumus untuk turunan vektor kecepatan yang digeneralisasi terhadap *H-frame* yang dapat dilihat pada Persamaan (2.49).

$$\dot{\xi} = M_H^{-1} (C_H(\xi)\dot{\xi} + G_H + O_H(\xi)) \Omega + E_H(\xi)\Omega^2 \quad (2.49)$$

2.1.4 Model Matematika [2]

Dari analisis kinematika dan dinamika diperoleh Persamaan model matematika dari *quadcopter* seperti pada Persamaan (2.50).

Dengan melihat secara sederhana pada Persamaan (2.50), posisi pada sumbu *Z*, dan posisi sudut *roll*, *pitch*, *yaw* dapat dikontrol secara langsung, berturut-turut dengan menggunakan U_1 , U_2 , U_3 , dan U_4 . Kontrol pada posisi maju (*X*), dan menyamping (*Y*) dapat dilakukan dengan mengatur sudut *pitch* dan *roll* dengan syarat gaya angkat (U_1) tidak sama dengan nol.

Model matematika ini nanti yang akan dijadikan acuan untuk merancang kontroler. Kontroler yang digunakan pada penelitian ini adalah kontroler PID (*Proportional Integral Derivative*) untuk gerak rotasi dan kontroler *Linear Quadratic Tracking* (LQT) untuk mengatur gerak translasi.

$$\begin{cases}
 \ddot{X} = \frac{U_1}{m} (\cos \Psi \sin \theta \cos \phi + \sin \Psi \sin \phi) \\
 \ddot{Y} = \frac{U_1}{m} (\sin \Psi \sin \theta \cos \phi - \cos \Psi \sin \phi) \\
 \ddot{Z} = -g + \frac{U_1}{m} (\cos \theta \cos \phi) \\
 \dot{p} = \frac{I_{yy} - I_{zz}}{I_{xx}} qr - \frac{I_r}{I_{xx}} q\Omega + \frac{U_2}{I_{xx}} \\
 \dot{q} = \frac{I_{zz} - I_{xx}}{I_{yy}} pr + \frac{I_r}{I_{yy}} q\Omega + \frac{U_2}{I_{yy}} \\
 \dot{r} = \frac{I_{xx} - I_{yy}}{I_{zz}} pq + \frac{U_4}{I_{zz}}
 \end{cases} \quad (2.50)$$

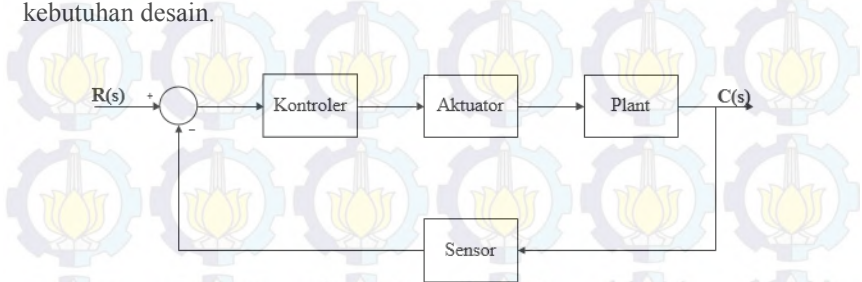
Nilai *input* dari *quadcopter* merupakan gaya angkat tiap *propeller*, yang telah dimodelkan secara teoritis dalam Persamaan (2.51), adalah sebagai berikut:

$$\begin{cases}
 U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\
 U_2 = bl(-\Omega_2^2 + \Omega_4^2) \\
 U_3 = bl(-\Omega_1^2 + \Omega_3^2) \\
 U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \\
 \Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4
 \end{cases} \quad (2.51)$$

2.2 Kontroler [3] [4]

Dalam sebuah sistem, kontroler memiliki kontribusi yang besar terhadap perilaku sistem. Pada prinsipnya hal itu disebabkan oleh tidak dapat diubahnya komponen penyusun sistem tersebut. Dengan artian bahwa karakteristik *plant* harus diterima sebagaimana adanya, sehingga perubahan perilaku sistem hanya dapat dilakukan melalui penambahan suatu *sub*-sistem, yaitu kontroler. Salah satu tugas komponen kontroler adalah meminimalkan sinyal kesalahan, yaitu perbedaan antara sinyal referensi dan sinyal aktual. Hal ini sesuai dengan tujuan sistem kontrol yaitu memperoleh sinyal aktual yang sama dengan sinyal referensi. Semakin cepat reaksi sistem (sinyal aktual) mengikuti sinyal referensi dan semakin kecil kesalahan yang terjadi, semakin baik kinerja sistem kontrol yang

diterapkan. Letak kontroler dapat dilihat pada blok diagram *closed loop* pada Gambar 2.10. Namun hal tersebut dapat bervariasi sesuai dengan kebutuhan desain.

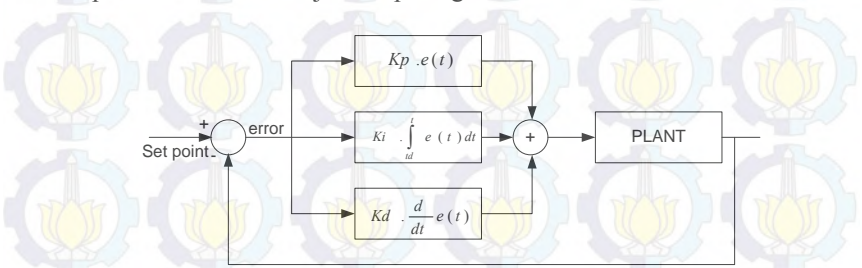


Gambar 2. 10 Blok Diagram *Closed Loop*

Pada penelitian ini digunakan kontroler *Proportional Integral Derivarive* (PID) yang digunakan untuk pengendalian gerak rotasi atau *inner loop*. Sedangkan untuk pengendalian gerak translasi atau *outer loop* digunakan kontroler *Linear Quadratic Tracking* (LQT).

2.2.1 *Proportional Integral Derivarive* (PID) [5]

Sebuah kontroler proporsional ditambah integral ditambah derivatif (PID) adalah salah satu mekanisme umpan balik yang banyak digunakan dalam sistem pengaturan industri. Sebuah kontroler PID menghitung nilai kesalahan sebagai perbedaan antara variabel proses terukur dan set point yang diinginkan. Hingga saat ini, PID dapat dikatakan sebagai kontroler terbaik karena tidak diperlukan teknik yang rumit dalam tahap perancangan. Namun, parameter PID yang digunakan dalam penghitungan harus diatur sesuai dengan sifat sistem. Diagram *loop* tertutup kontroler PID ditunjukkan pada gambar 2.11 berikut.



Gambar 2. 11 Pengaturan Loop Tertutup Kontroler PID

Algoritma penghitungan kontroler PID melibatkan tiga parameter yang terpisah, yaitu nilai-nilai proporsional, integral, dan derivatif, dinotasikan sebagai P, I dan D. Nilai P menentukan aksi kontrol berdasarkan perbandingan terhadap sinyal kesalahan. Nilai I memberikan aksi kontrol yang bersifat proporsional berdasarkan jumlah waktu saat terjadi kesalahan. Nilai D menentukan aksi kontrol yang bersifat proporsional terhadap perubahan kesalahan.

Secara heuristik, nilai-nilai ini dapat diklasifikasikan dari segi waktu terhadap sinyal kesalahan. Nilai P bergantung pada kesalahan saat ini. Nilai I bergantung pada kesalahan masa lalu. Nilai D bergantung pada prediksi kesalahan masa depan. Dengan penalaan ketiga konstanta ini, kontroler dapat memberikan aksi kontrol yang disesuaikan untuk kebutuhan proses tertentu. Respon kontroler dapat digambarkan sebagai respon terhadap kesalahan, sejauh mana kontroler dapat mencapai set point, serta derajat osilasi sistem. Walaupun demikian, penggunaan algoritma PID untuk pengaturan tidak menjamin pengaturan optimal dan stabilitas sistem. Beberapa aplikasi terkadang hanya memerlukan satu atau dua aksi kontrol untuk menyediakan pengaturan sistem yang tepat. Hal ini dicapai dengan menetapkan penguatan keluaran dari aksi kontrol yang tidak diinginkan ke nol. Oleh karena itu, sebuah kontroler PID dapat disebut sebagai kontroler PI, PD, P atau I tanpa adanya tindakan pengaturan masing-masing. [6]

Hubungan sinyal error dan sinyal kontrol pada kontroler tipe-PID standar dapat dinyatakan pada Persamaan 2.52:

$$u(t) = K_p \left[e(t) + \frac{1}{\tau_I} \int e(t) dt + \tau_D \frac{d}{dt} e(t) \right] \quad (2.52)$$

Atau dalam bentuk fungsi alih pada Persamaan 2.53:

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{\tau_I s} + \tau_D s \right)$$

$$\frac{U(s)}{E(s)} = \frac{K_p (\tau_I \tau_D s^2 + \tau_I s + 1)}{\tau_I s} \quad (2.53)$$

2.2.2 Linear Quadratic Tracking (LQT) [6]

LQT merupakan sistem pengatur linear yang *output* sistem mengikuti referensi (*trajectory*) yang diinginkan. Suatu sistem mempunyai persamaan *state* seperti Persamaan (2.54) vektor kesalahan seperti Persamaan (2.55).

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (2.54)$$

$$y(t) = Cx(t)$$

$$e(t) = z(t) - y(t) \quad (2.55)$$

Di mana $e(t)$ merupakan *error* yang merupakan selisih antara *input* referensi yang diberikan $z(t)$ dan *output* dari sistem $y(t)$. Indeks performansi didefinisikan pada Persamaan (2.56).

$$J = \frac{1}{2} e'(t_f)F(t_f)e(t_f) + \frac{1}{2} \int_{t_0}^{t_f} [e'(t)Q(t)e(t) + u'(t)R(t)u(t)] dt \quad (2.56)$$

Diasumsikan $F(t_f)$ dan $Q(t)$ adalah skalar atau matriks semidefinit positif yang simetris dengan dimensi $(m \times m)$, $R(t)$ adalah matriks definit positif yang juga simetris $(R \times R)$ matrix Q dan R adalah matriks pembobot yang menentukan kinerja dari sistem yang akan dikontrol.

Setelah mendapatkan model matematika sistem dalam bentuk *state-space* dan matriks $A(t)$ dan $B(t)$, matriks penyelesaian persamaan *differential Riccati* dapat didapatkan dengan Persamaan (2.57) untuk *finite-time case* dan Persamaan (2.58) untuk *infinite-time case*.

$$\dot{P}(t) = -P(t)A(t) - A'(t)P(t) + P(t)B(t)R^{-1}(t)B'(t)P(t) + C'(t)Q(t)C(t) \quad (2.57)$$

$$0 = -P(t)A(t) - A'(t)P(t) + P(t)B(t)R^{-1}(t)B'(t)P(t) + C'(t)Q(t)C(t) \quad (2.58)$$

Matriks Q dan R dapat diasumsikan sesuai dengan performansi yang diinginkan sistem. Setelah mendapatkan persamaan Riccati, persamaan diferensial vektor *non-homogen* dapat dicari menggunakan Persamaan (2.59).

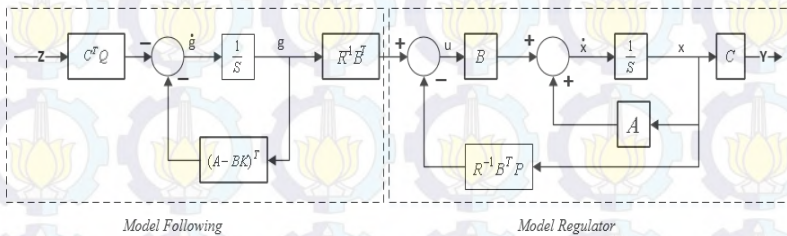
$$\dot{g}(t) = -[A(t) - B(t)R^{-1}B'(t)P(t)]'g(t) - C'(t)Q(t)z(t) \quad (2.59)$$

Setelah mendapatkan matriks $P(t)$ yang merupakan matriks definit positif yang simetris dan $g(t)$, nilai *gain feedback* $K(t)$, dan $u^*(t)$ dapat dicari dengan menggunakan Persamaan (2.60)-(2.61).

$$K(t) = R^{-1}(t)B'(t)P(t) \quad (2.60)$$

$$u^*(t) = -K(t)x^*(t) + R^{-1}(t)B'(t)g(t) \quad (2.61)$$

Persamaan diatas adalah persamaan diferensial dengan solusi aljabar Riccati. Dari solusi aljabar Riccati didapatkan nilai matriks K yang konstan yang tidak ada perubahan terhadap fungsi waktu. Blok diagram LQT ditunjukkan oleh Gambar 2.12 di mana terdiri dari 2 bagian, yaitu *model following* dan *model regulator*.



Gambar 2. 12 Blok Diagram LQT

2.3 Linearisasi [7]

Suatu sistem yang kompleks memiliki hubungan antara besaran-besaran fisik dalam sistem bersifat nonlinear. Untuk menyelesaikan persoalan nonlinear, dapat digunakan dengan menggunakan pendekatan persamaan linear. Salah satu karakteristik dari sistem nonlinear adalah ketergantungan perilaku respon sistem pada besaran dan *input* dari sistem.

Quadcopter merupakan sistem yang memiliki hubungan antar variable bersifat nonlinear. Pada Persamaan (2.27) terlihat bahwa persamaan tersebut terdapat hubungan antar variable bersifat nonlinear karena terdapat fungsi trigonometri. Untuk dapat melakukan proses kontrol yang menggunakan kontroler linear yaitu PID dan LQT, maka model matematika tersebut harus dilinearisasi. Teori linearisasi digunakan untuk mengatasi permasalahan sistem nonlinear agar dapat didekati dengan sistem linear.

Pada penelitian ini digunakan teknik linearisasi *feedback linearization* di mana merupakan salah satu teknik linearisasi dengan melakukan modifikasi pada variabel yang dikontrol sehingga dapat didekati dengan sistem linear. Misalkan suatu sistem mempunyai persamaan *state* seperti Persamaan (2.62).

$$\begin{aligned}\dot{x} &= f(x) + G(x)u \\ y &= h(x)\end{aligned}\tag{2.62}$$

Domain tersebut mengandung *origin*, sehingga akan didapatkan sinyal kontrol modifikasi pada Persamaan (2.63).

$$u = \alpha(x) + \beta(x)v\tag{2.63}$$

2.4 Ardupilot Mega 2.6 [3]

Mikrokontroler akan diprogram menggunakan bahasa C. Mikrokontroler akan dipadukan menjadi modul *Ardupilot Mega* pada Gambar 2.13. Spesifikasi dari mikrokontroler ATMega 2560 adalah:

- a. Memori : 256 KByte
- b. Frekuensi *Clock* : Maksimum 16Mhz
- c. I/O : 86 *port*
- d. ADC : 10 bit/16 *port*
- e. *Timer* 8 bit : 2 buah
- f. *Timer* 16 Bit : 4 buah
- g. Suplai tegangan : 4,5-5,5 VDC

Transmitter berada pada sisi pengguna yaitu pada *remote control* untuk memberikan nilai *input* kepada *quadcopter*, sinyal tersebut akan diterima *receiver* dan diteruskan ke mikrokontroler berupa pulsa. Tampilan *transmitter* dan *receiver* radio seperti pada Gambar 2.14. Spesifikasi dari radio *transmitter* adalah:

- a. Nama *Transmitter* : FrSky V8FR-II 2.4 Ghz 8 CH
- b. *Channel* : 8 buah
- c. Ukuran : 63,9 x 48,5 x 36,5 mm
- d. Arus : 50 mA
- e. Konsumsi daya : 60 mW

Disamping itu, spesifikasi untuk *receiver* adalah:

- a. Nama *receiver* : FrSky V8FR-II 2.4 Ghz 8 CH
- b. *Channel* : 8 buah
- c. Ukuran : 44 x 24 x 14 mm
- d. Berat : 9,3 gram
- e. *Range* tegangan : 3,0 – 16 V
- f. Arus : 30 mA
- g. *Range* : 1,5 – 2,5 km

2.6 Remote Control Turnigy [3]

Remote control merupakan salah satu perangkat yang digunakan untuk mode penerbangan manual ditunjukkan oleh Gambar 2.15.



Gambar 2. 15 Remote Control (*Transmitter*) Turnigy 9XR

Remote control tersebut disambungkan dengan dengan radio *transmitter* sehingga dapat mengirimkan data pada mikrokontroler dan diterima oleh *radio receiver*. Spesifikasi dari *remote control* adalah

- a. Nama : Turnigy 9XR Radio *Transmitter*
- b. *Channel* : 24 buah
- c. *Display* : 128 x 64 LCD
- d. *Battery* : 3 *cell* lipo
- e. *Stick mode* : 1, 2, 3, 4

2.7 *Electronic Speed Controller (ESC)* [3]

ESC merupakan *driver* dari motor *brushless* DC. ESC di-*trigger* oleh sinyal PWM yang dikendalikan oleh mikrokontroler ATmega 2560 ditunjukkan oleh Gambar 2.16. Sinyal PWM tersebut akan men-*drive* motor dengan kecepatan yang linear dengan besar pulsa yang diberikan. Spesifikasi dari ESC adalah:

- a. Nama : ESC TBS Bulletproof 30 *Ampere*
- b. *Battery* : 2 – 4 *cell*
- c. Arus Maksimum : 30 A
- d. Tegangan : 6,4 – 16,8 V



Gambar 2. 16 ESC TBS Bulletproof 30 *Ampere*

2.8 *Motor BLDC dan Propeller* [3]

Motor yang digunakan sebagai aktuator adalah motor DC tanpa sikat SunnySky X2212 KV980 yang sumbuinya dipasang *propeller* berukuran 10 cm x 4,5 cm. Bentuk motor DC tanpa sikat dan *propeller* terlihat seperti pada Gambar 2.17.



Gambar 2. 17 Motor *Quadcopter* dan *Propeller*

2.9 *Ground Station* [3]

Untuk dapat mengetahui data penerbangan dari *quadcopter* dapat digunakan *Ground station*. Data dari mikrokontroler dikirimkan secara *wireless* menuju komputer *ground station* dengan komunikasi serial menggunakan *telemetry radio 433 Mhz*. Data serial langsung dibaca pada *software Mission Planner* pada *ground station* yang ditunjukkan oleh Gambar 2.18.



Gambar 2. 18 *Telemetry* dan *Ground Station*

BAB III

PERANCANGAN SISTEM

Pada Bab ini membahas mengenai perancangan sistem kontrol *quadcopter* pada gerak *cruise*. Perancangan sistem terdiri dari spesifikasi sistem, identifikasi kebutuhan, perancang mekanik, perancangan elektronik, dan perancangan kontrol PID dan LQT.

3.1 Spesifikasi Sistem

Pada Tugas Akhir ini menggunakan *quadcopter* rakitan yang memiliki 6 dof (*degree of freedom*), yang terdiri dari 3 dof rotasi dan 3 dof translasi yang akan dibuat menjadi spesifikasi sistem tertentu secara *hardware* maupun simulasi pada matlab dapat diuraikan sebagai berikut:

- a. *Quadcopter* dapat bergerak rotasi maksimal 5,7 derajat
- b. *Quadcopter* dapat digerakkan secara manual dengan menggunakan *remote control*
- c. *Quadcopter* dapat melakukan gerak *cruise* sesuai dengan referensi yang diberikan.

3.2 Identifikasi Kebutuhan

Quadcopter yang digunakan harus mampu memenuhi spesifikasi yang ada. Untuk memenuhi spesifikasi sistem tersebut maka terdapat kebutuhan yang harus dipenuhi antara lain:

- a. *Quadcopter* harus dibuat simetris, seimbang, dan ringan untuk menghindari kelembaman yang sangat besar.
- b. Baterai yang digunakan minimal dapat membuat *quadcopter* terbang sekitar 15 menit.
- c. *Remote control* minimal terdapat 4 kanal sebagai pergerakan naik/turun, *roll*, *pitch*, dan *yaw*

3.3 Perancangan Mekanik

Sistem mekanik yang baik akan mendukung pergerakan *quadcopter* menjadi lebih baik. Untuk mendukung pergerakan tersebut, perancangan mekanik harus dibuat simetris, seimbang, dan ringan. Dalam hal ini *Center of Gravity* (COG) dari *quadcopter* harus diperhatikan. Berat dari keempat sisi *quadcopter* juga harus sama untuk memenuhi kriteria COG. Selain itu juga posisi baterai harus terletak tepat di tengah titik massa *quadcopter* agar

seimbang. Beban yang diberikan kepada *quadcopter* tidak boleh melebihi daya angkat dari *quadcopter* agar dapat terbang.

Pada penelitian Tugas Akhir ini, desain dari *quadcopter* yang telah dirancang ditunjukkan pada Gambar 3.1. *Quadcopter* tersebut merupakan *quadcopter* rakitan/*hobby* yang terdapat pada Lab B204 Teknik Pengaturan.

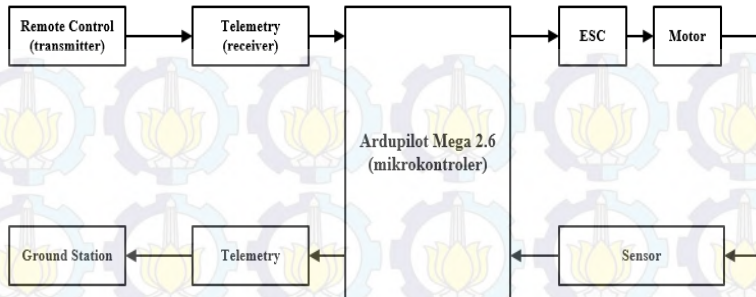


Gambar 3. 1 Desain Mekanik *Quadcopter* yang Telah Dirancang

3.4 Perancangan Elektronik [3]

Perancangan elektronika yang ada pada *quadcopter* terdiri dari sistem kontroler yang berupa Ardupilot Mega 2.6 di mana telah sudah dilengkapi beberapa sensor dan *port* yang memudahkan untuk penggunaan sebagai komunikasi data.

Board Ardupilot Mega 2.6 ini dilengkapi dengan ATmega 2560 sebagai mikrokontroler dari *flight controller*. Selain itu juga ditambahkan GPS yang digunakan untuk mengetahui posisi dari *quadcopter*, *telemetry* 433 Mhz yang digunakan sebagai komunikasi data antara kontroler Ardupilot Mega 2.6 dengan *ground station* yang menggunakan *software Mission Planner*. Perancangan keseluruhan dari sistem *quadcopter* ditunjukkan pada Gambar 3.2.



Gambar 3. 2 Perancangan Sistem Pengendalian *Quadcopter*

3.5 Pemodelan Motor dan *Propeller*

Untuk mengetahui model linear dari *quadcopter*, maka dilakukan beberapa pengukuran, yaitu pengukuran untuk mengetahui model linear dari hubungan pulsa dengan kecepatan motor (rpm dan rad/s) dan pengukuran untuk mengetahui hubungan antara pulsa dengan gaya angkat motor (kg).

3.5.1 Pengukuran Kecepatan Motor [3]

Pengukuran ini dilakukan dengan menggunakan tachometer laser dan menggunakan sinyal masukan berupa PWM dari Arduino.

Dari hasil pengukuran diperoleh persamaan linear hubungan pulsa motor terhadap kecepatan (rad/s) pada masing-masing motor terdapat pada Persamaan (3.1).

$$\begin{aligned}
 y_1 &= 44,851x - 1760,5 \\
 y_2 &= 44,10x - 1603,7 \\
 y_3 &= 44,8x - 1670,4 \\
 y_4 &= 45,047x - 1618,7
 \end{aligned}
 \tag{3.1}$$

3.5.2 Pengukuran Gaya Angkat Motor [3]

Langkah-langkah dalam pengukuran gaya angkat motor adalah dengan meletakkan motor dan *propeller* tepat diatas beban sehingga memiliki berat total 1,618 kg. Beban tersebut merupakan botol aqua 1,5 liter dengan diisi air. Kemudian diletakkan diatas timbangan digital dan motor dijalankan dengan memberikan pulsa atau sinyal PWM dari arduino. Hasil pembacaan berat pada timbangan dikurangi dengan beban 1,618 kg adalah gaya angkat pada masing-masing motor. Pada eksperimen ini dilakukan pengukuran gaya angkat pada masing-masing motor *propeller*.

Dari pengukuran diperoleh persamaan linear hubungan pulsa motor terhadap gaya angkat kg pada masing-masing motor terdapat pada Persamaan (3.2).

$$\begin{aligned}y_1 &= 0,0076x - 0,6097 \\y_2 &= 0,0076x - 0,5968 \\y_3 &= 0,0076x - 0,5968 \\y_4 &= 0,0078x - 0,6026\end{aligned}\tag{3.2}$$

3.6 Identifikasi dan Validasi Sistem [3]

Model matematika dari *quadcopter* pada bab sebelumnya diketahui bahwa terdapat beberapa parameter yang harus dicari nilainya. parameter tersebut ialah sebagai berikut:

- a) Massa
- b) Panjang lengan *quadcopter*
- c) Momen inersia
- d) Pemetaan gaya angkat terhadap pulsa
- e) Konstanta *thrust*
- f) Konstanta *drag*.

Massa dari sistem keseluruhan diukur menggunakan timbangan digital. Momen inersia dari sistem pada sumbu X, sumbu Y, dan sumbu Z diperoleh menggunakan identifikasi parametrik dengan cara menerbangkan *quadcopter* lalu didapatkan data keluaran dari sensor yaitu berupa kecepatan sudut, serta nilai persentase *throttle* dari *remote control*.

Sedangkan, untuk mendapatkan nilai konstanta *thrust* dan konstanta *drag*, dilakukan pengujian pada masing-masing motor dengan diberi sinyal

PWM, lalu diukur kecepatan dari keempat motor tersebut, dan dihitung konstanta *thrust* dan *drag* dari *quadcopter*.

Untuk mendapatkan konstanta *thrust*, *quadcopter* diterbangkan dalam posisi *hover* karena pada posisi tersebut, gaya yang dihasilkan oleh motor untuk mengangkat *quadcopter* sama dengan gaya yang mendorong *quadcopter* menuju *ground*. Dengan asumsi gaya gravitasi tetap, maka konstanta *thrust* dapat dihitung menggunakan Persamaan (3.3).

$$b = \frac{m \cdot g}{\sum_{i=1}^4 \Omega_i^2} \quad (3.3)$$

Konstanta *drag* dihitung dengan persamaan gerak lurus berubah beraturan. Pengukuran konstanta *drag* dilakukan dengan mengambil data penerbangan *quadcopter* pada saat *take-off*. Gaya yang terjadi saat *quadcopter* bergerak ke atas dapat diperoleh menggunakan Persamaan (3.4).

$$\sum F = Thrust - mg - Drag \quad (3.4)$$

$$ma = (\tau_1 + \tau_2 + \tau_3 + \tau_4) - mg - D \quad (3.5)$$

Jika Persamaan (3.5) disubstitusikan pada Persamaan (3.4) maka Persamaan (3.4) menjadi Persamaan (3.6).

$$D = ma - ((\tau_1 + \tau_2 + \tau_3 + \tau_4) - mg) \quad (3.6)$$

Di mana percepatan dan konstanta *drag* dari *quadcopter* dapat dihitung menggunakan Persamaan (3.7)-(3.8).

$$a = \frac{s - v_0 t}{0,5 t^2} \quad (3.7)$$

$$CD = \frac{D}{\Omega^2} \quad (3.8)$$

Dengan ketentuan sebagai berikut:

CD = Konstanta *Drag*

v_0 = Kecepatan awal (m/s)

t = Waktu (s)

a = Percepatan (m/s^2)

s = Jarak yang ditempuh (m)

τ_i = Gaya angkat, $i=1,2,3,4$

D = Gaya *drag*

Ω = Kecepatan motor

Parameter hasil identifikasi menggunakan pengukuran terdapat pada Tabel 3.1.

Tabel 3. 1 Parameter *Quadcopter* Hasil Pengukuran

No	Parameter	Nilai	Satuan
1	Massa <i>quadcopter</i> (m)	1,26	kg
2	Jari-jari <i>quadcopter</i> (l)	0,206	meter
3	Konstanta <i>Thrust</i>	$1,68918 \times 10^{-5}$	$N.sec^2$
4	Konstanta <i>Drag</i>	$4,19 \times 10^{-6}$	$Nm.sec^2$

Pada model matematika rotasi *quadcopter*, terdapat beberapa parameter yang belum diketahui. Untuk mendapatkannya dapat dicari menggunakan Pendekatan Penyelesaian Persamaan Linear Simultan (P3LS) dengan memodifikasi model rotasi *quadcopter* menjadi Persamaan (3.9)-(3.11).

$$\dot{p} = a_1 qr + b_1 q\Omega + c_1 U_2 \quad (3.9)$$

$$\dot{q} = a_2 pr + b_2 p\Omega + c_2 U_3 \quad (3.10)$$

$$\dot{r} = a_3 pq + b_3 U_4 \quad (3.11)$$

Dimana data parameter tersebut diperoleh dari data penerbangan. Dari Persamaan (3.9) dapat dibuat suatu persamaan matriks pada Persamaan (3.12) dan nilai a_1, b_1, c_1 dapat dicari nilainya dengan menggunakan Persamaan (3.13).

$$\begin{bmatrix} qr & q\Omega & U_2 \\ b_1 \end{bmatrix} = \dot{p} \quad (3.12)$$

$$\begin{bmatrix} a_1 \\ b_1 \\ c_1 \end{bmatrix} = \left(\begin{bmatrix} qr \\ q\Omega \\ U_2 \end{bmatrix} \begin{bmatrix} qr & q\Omega & U_2 \end{bmatrix} \right)^{-1} \begin{bmatrix} qr \\ q\Omega \\ U_2 \end{bmatrix} \dot{p} \quad (3.13)$$

Persamaan (3.10) dapat dibuat suatu persamaan matriks pada Persamaan (3.14) dan nilai a_2 , b_2 , c_2 dapat dicari nilainya dengan menggunakan Persamaan (3.15).

$$\begin{bmatrix} pr & p\Omega & U_3 \\ b_2 \\ c_2 \end{bmatrix} = \dot{q} \quad (3.14)$$

$$\begin{bmatrix} a_2 \\ b_2 \\ c_2 \end{bmatrix} = \left(\begin{bmatrix} pr \\ p\Omega \\ U_3 \end{bmatrix} \begin{bmatrix} pr & p\Omega & U_3 \end{bmatrix} \right)^{-1} \begin{bmatrix} pr \\ p\Omega \\ U_3 \end{bmatrix} \dot{q} \quad (3.15)$$

Persamaan (3.11) dapat dibuat suatu persamaan matriks pada Persamaan (3.16) dan nilai a_3 dan b_3 dapat dicari dengan menggunakan Persamaan (3.17).

$$\begin{bmatrix} pq & U_4 \\ b_3 \end{bmatrix} = \dot{r} \quad (3.16)$$

$$\begin{bmatrix} a_3 \\ b_3 \end{bmatrix} = \left(\begin{bmatrix} pq \\ U_4 \end{bmatrix} \begin{bmatrix} pq & U_4 \end{bmatrix} \right)^{-1} \begin{bmatrix} pq \\ U_4 \end{bmatrix} \dot{r} \quad (3.17)$$

Dari hasil Persamaan (3.13)-(3.17) didapatkan nilai parameter yang ditunjukkan pada Tabel (3.2).

Tabel 3. 2 Parameter *Quadcopter* Hasil Identifikasi Parametrik

No	Konstanta	Nilai
1	a_{11}	-0,5495
2	b_{11}	-0,0017
3	c_{11}	0,2052
4	a_{21}	0,1675
5	b_{21}	-0,0094
6	c_{21}	2,955
7	a_{31}	-2,0257
8	b_{31}	0,0594

Identifikasi model *quadcopter* juga dilakukan dengan menggunakan identifikasi fisik, berupa penghitungan momen inersia dan konstanta motor *propeller*. Penghitungan identifikasi fisik terdapat pada lampiran A3.

Hasil dari identifikasi fisik terdapat terdapat pada Tabel 3.3.

Tabel 3. 3 Parameter *Quadcopter* Hasil Identifikasi Fisik

No	Konstanta	Nilai
1	a_{11}	0,25327
2	b_{11}	-0,0198
3	c_{11}	595,174
4	a_{21}	-0,2537
5	b_{21}	0,0198
6	c_{21}	797,04
7	a_{31}	0
8	b_{31}	595,174

Selanjutnya akan dilakukan proses validasi untuk membandingkan hasil identifikasi fisik dan parametrik dengan menggunakan Persamaan (3.9)-(3.10) di mana telah diketahui nilai parameter hasil penerbangan.

Hasil proses validasi dengan menggunakan RMSE sebagai pembanding terdapat pada Tabel 3.4.

Tabel 3.4 Validasi Hasil Identifikasi Parametrik dan Identifikasi Fisik

Besaran	RMSE Identifikasi Parametrik	RMSE Identifikasi Fisik
\dot{p}	3,2232%	190,34%
\dot{q}	1,3507%	177,53%
\dot{r}	5,3405%	92,525%

3.7 Model Matematika Hasil Identifikasi [3]

Setelah diperoleh parameter dari sistem, maka dapat dituliskan kembali model matematika dari sistem *quadcopter* pada Persamaan (3.18)-(3.23).

$$\ddot{X} = \frac{U_1}{1,26} (\cos \Psi \sin \theta \cos \phi + \sin \Psi \sin \phi) \quad (3.18)$$

$$\ddot{Y} = \frac{U_1}{1,26} (\sin \Psi \sin \theta \cos \phi - \cos \Psi \sin \phi) \quad (3.19)$$

$$\ddot{Z} = -9,81 + \frac{U_1}{1,26} (\cos \theta \cos \phi) \quad (3.20)$$

$$\dot{p} = -0,5495 qr - 0,00017 q\Omega + 0,2052 U_2 \quad (3.21)$$

$$\dot{q} = 0,1675 pr - 0,0094 p\Omega + 2,955 U_3 \quad (3.22)$$

$$\dot{r} = -2,0257 pq + 0,0954 U_4 \quad (3.23)$$

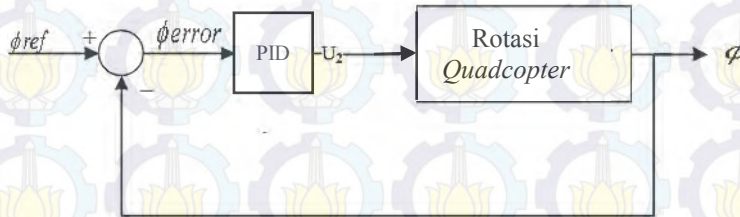
3.8 Perancangan Kontroler PID [8]

PID digunakan sebagai kontroler *quadcopter* pada bagian rotasi. Dimana gerak rotasi *quadcopter* terdiri dari gerak *roll*, *pitch*, dan *yaw*. Untuk mendesain kontroler PID diperlukan linearisasi, karena *quadcopter* adalah *plant* nonlinear. Metode linearisasi yang digunakan adalah *feedback linearization*. Untuk tuning parameter Kp, Ki, Kd pada PID digunakan metode *procedural tuning* atau dengan kata lain dengan metode konseptual. Maksud dari cara ini adalah untuk mencari parameter PID dilakukan pencarian nilai Kp hingga respon mendekati *reference* yang diberikan dan tidak ada osilasi. Selanjutnya, dicari nilai Ki, untuk mendekati nilai

respon menuju *reference*. Namun, dengan adanya nilai K_i , ini dapat berakibat munculnya *overshoot*, maka dicari nilai K_d untuk menghilangkan *overshoot* tersebut^[8].

3.8.1 Perancangan Kontroler PID Sudut Roll

Pada perancangan kontroler sudut *roll* digunakan Persamaan (3.21). Sudut *roll* berarti akan mempengaruhi pergerakan *quadcopter* pada sumbu Y. Blok diagram pengontrolan sudut *roll* ditunjukkan pada Gambar 3.3 dengan ϕ_{ref} adalah sinyal kontrol dari kontroler translasi sumbu Y. Hal tersebut dikarenakan gerak translasi dikontrol secara *cascade*.



Gambar 3. 3 Blok Diagram Pengendalian Sudut Roll

Untuk perancangan PID sudut *roll*, langkah pertama ialah mendefinisikan hubungan linear antar *variable* pada model matematika sudut *roll*. Hubungan linear tersebut terdapat pada Persamaan (3.24)-(3.26).

$$\dot{\phi} = p \quad (3.24)$$

$$\ddot{\phi} = \dot{p} \quad (3.25)$$

$$\ddot{\phi} = -0,5495 q r - 0,00017 q \Omega + 0,2052 U_2 \quad (3.26)$$

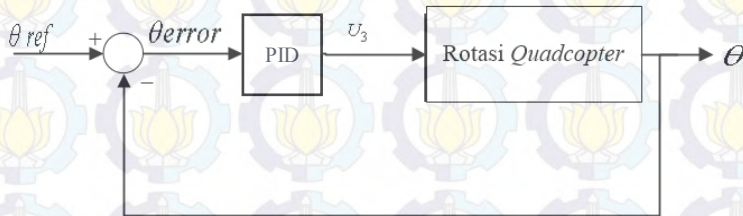
Menggunakan kontroler PID, berarti harus mencari nilai K_p , K_i , dan K_d . Pada pengaturan gerak rotasi *quadcopter* ini digunakan cara *procedural tuning* dan didapatkan nilai K_P , K_i , dan K_d untuk pengaturan sudut *roll* pada Tabel 3.5.

Tabel 3. 5 Nilai K_p , K_i , dan K_d Kontroler PID Sudut Roll

No	Parameter	Nilai
1	K_p Roll	800
2	K_i Roll	0,01
3	K_d Roll	7000

Dari Tabel 3.5 dapat dilihat bahwa nilai K_p bernilai besar. Hal dikarenakan dibutuhkan respon yang cepat untuk *quadcopter*. Respon rotasi harus lebih cepat dibanding respon translasi. Dikarenakan nilai K_p yang besar, selain respon yang semakin cepat, maka *overshot* yang dihasilkan juga semakin besar. Untuk mengantisipasi hal tersebut maka nilai K_d juga harus besar.

3.8.2 Perancangan Kontroler PID Sudut *Pitch*



Gambar 3. 4 Blok Diagram Pengendalian Sudut *Pitch*

Pengontrolan gerak rotasi sudut *pitch* menggunakan Persamaan (3.22). Sudut *pitch* akan mengakibatkan *quadcopter* bergerak pada sumbu X. Blok diagram pengendalian sudut *pitch* ditunjukkan oleh Gambar 3.4 di mana θ_{ref} adalah sinyal kontrol dari kontroler translasi pada sumbu X.

Hubungan antar *variable* pada model matematika gerak rotasi sudut *pitch* yang terdapat pada Persamaan (3.27)-(3.29).

$$\dot{\theta} = q \tag{3.27}$$

$$\ddot{\theta} = \dot{q} \tag{3.28}$$

$$\ddot{\theta} = 0,1675 p_r - 0,0094 p_\Omega + 2,955 U_3 \tag{3.29}$$

Penentuan nilai K_p , K_i , dan K_d untuk pengaturan sudut *pitch* ini digunakan cara *procedural tuning*. Didapatkan nilai K_p , K_i , dan K_d untuk kontroler PID sudut *pitch* pada Tabel 3.6.

Tabel 3. 6 Nilai K_p , K_i , dan K_d Kontroler PID Sudut *Pitch*

No	Parameter	Nilai
1	K_p <i>Pitch</i>	100
2	K_i <i>Pitch</i>	0,01
3	K_d <i>Pitch</i>	1200

3.8.3 Perancangan Kontroler PID Sudut *Yaw*

Perancangan PID pada pengendalian sudut *yaw* menggunakan Persamaan (3.23).

Kontroler sudut *yaw* dirancang sebagai regulator, di mana sudut *yaw* akan dipertahankan pada sekitar 0 radian.

Hubungan antar *variable* pada model matematika gerak rotasi sudut *yaw* terdapat pada Persamaan (3.30)-(3.32)

$$\dot{\psi} = r \tag{3.30}$$

$$\dot{\psi} = \dot{r} \tag{3.31}$$

$$\ddot{\psi} = -2,0257 pq + 0,0954 U_4 \tag{3.32}$$

Penentuan nilai K_p , K_i , dan K_d untuk pengaturan sudut *yaw* ini digunakan cara *procedural tuning error*. Didapatkan nilai K_p , K_i , dan K_d untuk kontroler PID sudut *pitch* pada Tabel 3.7.

Tabel 3. 7 Nilai K_p , K_i , dan K_d Kontroler PID Sudut *Yaw*

No	Parameter	Nilai
1	$K_p \text{ Yaw}$	50
2	$K_i \text{ Yaw}$	0,0001
3	$K_d \text{ Yaw}$	950

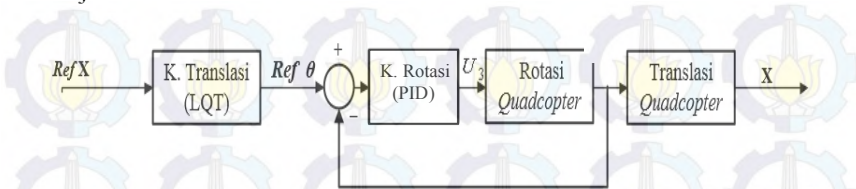
3.9 Perancangan *Linear Quadratic Tracking* (LQT)

LQT digunakan untuk pengendalian gerak translasi. Gerak translasi tersebut terdiri dari 3 macam, yaitu gerak translasi pada sumbu X, gerak translasi pada sumbu Y, dan gerak translasi pada sumbu Z. Sama halnya dengan perancangan PID, perancangan LQT juga memerlukan linearisasi karena *quadcopter* merupakan sistem nonlinear. Perancangan LQT ini menggunakan *feedback linearization* yang digunakan untuk menghilangkan pengaruh nonlinear dari sistem tersebut.

3.9.1 Perancangan Kontroler LQT Pengendalian Sumbu X

Untuk dapat mengendalikan sumbu X, maka *quadcopter* harus dalam keadaan stabil sehingga dapat melakukan perubahan sudut *pitch*. Oleh karena itu perancangan gerak translasi digunakan metode pengendalian secara *cascade* antara kontroler gerak translasi dan kontroler

gerak rotasi. Blok diagram pengendalian gerak translasi sumbu X ditunjukkan oleh Gambar 3.5.



Gambar 3.5 Blok Diagram Pengendalian Gerak Translasi *Quadcopter* pada Posisi X

Pada perancangan LQT untuk posisi sumbu X digunakan Persamaan (3.18). Persamaan tersebut memiliki tiga *variable* ϕ , θ , ψ di mana gerak pada sumbu X bergantung pada besarnya nilai sudut θ dan juga mendapat pengaruh secara langsung oleh sudut ϕ . Dimisalkan bahwa *quadcopter* bersifat *rigid* untuk sudut ψ sehingga besarnya perubahan sudut *yaw* bernilai sangat kecil atau $\psi \approx 0$.

Dari permisalan tersebut, Persamaan (3.18) akan menjadi Persamaan (3.33).

$$\ddot{X} = (\sin \theta \cos \phi) \frac{U_1}{1,26} \quad (3.33)$$

Untuk menghilangkan efek nonlinear dari sistem, maka sinyal kontrol harus dimodifikasi pada Persamaan (3.34) dan persamaan *state* sistem menjadi Persamaan (3.35).

$$U_x^* = (\sin \theta \cos \phi) \frac{U_1}{1,26} \quad (3.34)$$

$$\ddot{X} = U_x^* \quad (3.35)$$

Karena sinyal kontrol LQT merupakan θ_{ref} maka dilakukan konversi sinyal kontrol U_x^* menjadi θ pada Persamaan (3.37).

$$\sin \theta = \frac{(1,26)U_x^*}{U_1 \cos \phi} \quad (3.36)$$

$$\theta = \arcsin\left(\frac{(1,26)U_x^*}{U_1 \cos \phi}\right) \quad (3.37)$$

State space dari Persamaan (3.37) terdapat pada Persamaan (3.38)-(3.39).

$$\begin{bmatrix} \dot{X} \\ \ddot{X} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} X \\ \dot{X} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} U_x^* \quad (3.38)$$

$$Y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ \dot{X} \end{bmatrix} \quad (3.39)$$

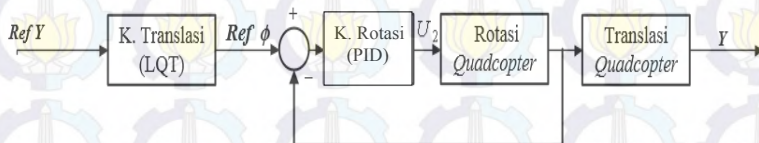
Dari Persamaan (3.38)-(3.39) didapatkan matriks A, B, dan C pada Persamaan (3.67).

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (3.40)$$

Pada perancangan kontroler LQT ini, digunakan satu kontroler saja untuk mengatur sumbu X, Y, dan Z atau disebut dengan *dependent controller*, sehingga untuk nilai Q dan R hanya ada satu saja untuk semua sumbu, dan *tuning* nilai Q dan R dilakukan dengan cara *try and error*.

3.9.2 Perancangan Kontroler LQT Pengendalian Sumbu Y

Gerak translasi sumbu Y dapat terjadi karena adanya gerak rotasi sudut *roll*. Sistem yang digunakan untuk gerak translasi sumbu Y juga menggunakan sistem *cascade* yaitu seperti pada Gambar 3.6.



Gambar 3. 6 Blok Diagram Pengendalian Gerak Translasi *Quadcopter* pada Posisi Y

Pada perancangan LQT untuk posisi sumbu Y digunakan Persamaan (3.19). Persamaan tersebut memiliki tiga variabel ϕ , θ , ψ di mana gerak

pada sumbu Y bergantung pada besarnya nilai sudut ϕ dan juga mendapat pengaruh secara langsung oleh sudut θ . Dimisalkan bahwa *quadcopter* bersifat *rigid* untuk sudut ψ sehingga besarnya perubahan sudut *yaw* bernilai sangat kecil atau $\psi \approx 0$.

Dari permisalan tersebut, Persamaan (3.19) akan menjadi Persamaan (3.41).

$$\ddot{Y} = -(\sin \phi) \frac{U_1}{1,26} \quad (3.41)$$

Untuk menghilangkan efek nonlinear dari sistem, maka sinyal kontrol harus dimodifikasi pada Persamaan (3.42) dan persamaan *state* sistem menjadi Persamaan (3.43)

$$U_y^* = -(\sin \phi) \frac{U_1}{1,26} \quad (3.42)$$

$$\ddot{Y} = U_y^* \quad (3.43)$$

Karena sinyal kontrol LQT merupakan ϕ_{ref} maka dilakukan konversi sinyal kontrol U_y^* menjadi ϕ pada Persamaan (3.45).

$$\sin \phi = -\frac{(1,26)U_y^*}{U_1} \quad (3.44)$$

$$\phi = \arcsin \left(-\frac{(1,26)U_y^*}{U_1} \right) \quad (3.45)$$

State space dari Persamaan (3.44) terdapat pada Persamaan (3.46)-(3.47).

$$\begin{bmatrix} \dot{Y} \\ \ddot{Y} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Y \\ \dot{Y} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} U_y^* \quad (3.46)$$

$$Y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} Y \\ \dot{Y} \end{bmatrix} \quad (3.47)$$

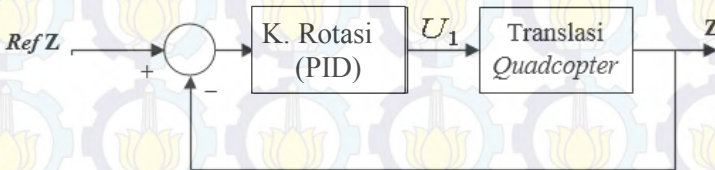
Dari Persamaan (3.45)-(3.46) didapatkan matriks A, B, dan C pada Persamaan (3.48).

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad C = [1 \quad 0] \quad (3.48)$$

Pada perancangan kontroler LQT ini, digunakan satu kontroler saja untuk mengatur sumbu X, Y, dan Z atau disebut dengan *dependent controller*, sehingga untuk nilai Q dan R hanya ada satu saja untuk semua sumbu, dan *tuning* nilai Q dan R dilakukan dengan cara *try and error*.

3.9.3 Perancangan Kontroler LQT Pengendalian Sumbu Z

Pada perancangan LQT sumbu Z, untuk dapat bergerak *take off* tidak memerlukan gerak rotasi. Blok diagram LQT pada sumbu Z ditunjukkan pada Gambar 3.7.



Gambar 3. 7 Blok Diagram Pengendalian Gerak Translasi *Quadcopter* pada Sumbu Z

Perancangan LQT sumbu Z menggunakan Persamaan (3.20). Untuk menghilangkan efek nonlinear dari sistem, maka sinyal kontrol harus dimodifikasi pada Persamaan (3.50) dan persamaan *state* sistem menjadi Persamaan (3.51).

$$\ddot{Z} = -9,81 + (\cos \theta \cos \phi) \frac{U_1}{1,26} \quad (3.49)$$

$$U_1^* = -9,81 + (\cos \theta \cos \phi) \frac{U_1}{1,26} \quad (3.50)$$

$$\ddot{Z} = U_1^* \quad (3.51)$$

Persamaan (3.51) dapat dibuat *state space* pada Persamaan (3.52)-(3.53).

$$\begin{bmatrix} \dot{Z} \\ \ddot{Z} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Z \\ \dot{Z} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} U_1^* \quad (3.52)$$

$$Y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} Z \\ \dot{Z} \end{bmatrix} \quad (3.53)$$

Persamaan (3.52)-(3.53) didapatkan matriks A, B, dan C pada Persamaan (3.54).

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (3.54)$$

Pada perancangan kontroler LQT ini, digunakan satu kontroler saja untuk mengatur sumbu X, Y, dan Z atau disebut dengan *dependent controller*, sehingga untuk nilai Q dan R hanya ada satu saja untuk semua sumbu, dan *tuning* nilai Q dan R dilakukan dengan cara *try and error*. Parameter Q dan R untuk kontroler LQT sumbu X, Y, dan Z yang dihasilkan dari cara *try and error* ditampilkan pada Tabel 3.8

Tabel 3.8 Nilai Q dan R Kontroler LQT Pengendalian *Quadcopter* pada Sumbu X, Y, dan Z

No	Parameter	Nilai
1	Q	100
2	R	0,1



BAB IV PENGUJIAN DAN ANALISIS

Berdasarkan spesifikasi dan perancangan sistem yang telah dijelaskan pada bab sebelumnya, maka dilakukan pengujian dan analisis terhadap perilaku sistem. Tujuan dari Pengujian adalah untuk mengetahui performansi sistem hasil perancangan. Jika performansi yang diharapkan belum sesuai, maka perlu diadakan analisis untuk penyempurnaan kinerja sistem sekaligus dapat digunakan untuk pengembangan lebih lanjut.

Pada bab ini dipaparkan mengenai hasil simulasi dari perancangan gerak lateral *quadcopter* menggunakan kontroler LQT di mana parameter dari kontroler akan didapatkan dengan *tuning* menggunakan *try and error*.

4.1 Simulasi Pengujian PID dengan Nilai Kp Berbeda

Perancangan kontroler PID menggunakan parameter Kp, Ki, dan Kd. Setiap parameter mempunyai fungsinya masing-masing. Dalam pengujian ini dilakukan uji respon sudut *roll* dan *pitch* dengan nilai Kp yang berbeda. Nilai Kp ini mempengaruhi cepatnya respon menuju kondisi *steady state*.

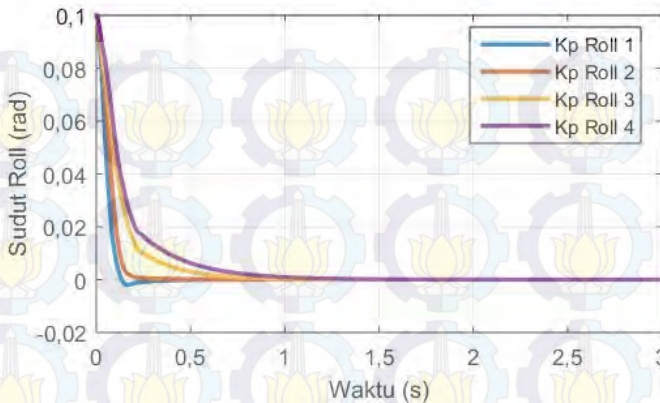
4.1.1 Simulasi Pengujian PID Pengendalian Sudut *Roll* dengan Nilai Kp Berbeda

Pada pengujian ini ditetapkan nilai Ki *roll* ditetapkan sebesar 0,01 dan juga nilai Kd *roll* sebesar 7000. Penentuan nilai parameter PID ini dilakukan dengan cara *procedural tuning*. Nilai Kp yang diuji ada 4 yang seperti pada Tabel 4.1

Tabel 4.1 Variasi Nilai Kp pada PID Pengendalian Sudut *Roll*

No	<i>Roll</i>	Nilai Kp
1	Kp <i>Roll</i> 1	800
2	Kp <i>Roll</i> 2	600
3	Kp <i>Roll</i> 3	500
4	Kp <i>Roll</i> 4	400

Hasil respon dari variasi nilai Kp ditunjukkan pada Gambar 4.1 spesifikasi dari respon ditunjukkan pada Tabel 4.2



Gambar 4. 1 Respon Variasi Nilai Kp pada Pengendalian PID Sudut Roll

Tabel 4. 2 Karakteristik Respon Variasi Nilai Kp pada PID Pengendalian Sudut Roll

Variasi Kp	Time Constant (s)	Settling Time (s)	Rise Time (s)
KP Roll 1	0,13	0,39	0,38
KP Roll 2	0,19	0,57	0,56
KP Roll 3	0,24	0,72	0,71
KP Roll 4	0,28	0,84	0,82

Ket: t_s ($\pm 5\%$), t_r (5%-95%)

Dari respon variasi nilai Kp pada sudut roll sesuai dengan Gambar 4.1 dan Tabel 4.2 terlihat bahwa nilai Kp Roll 1 memiliki respon paling cepat menuju *steady state*. Sudut roll yang dibutuhkan pada *plant quadcopter* ini diusahakan mempunyai respon rotasi yang cepat. Oleh karena itu nilai Kp yang dipilih adalah 100.

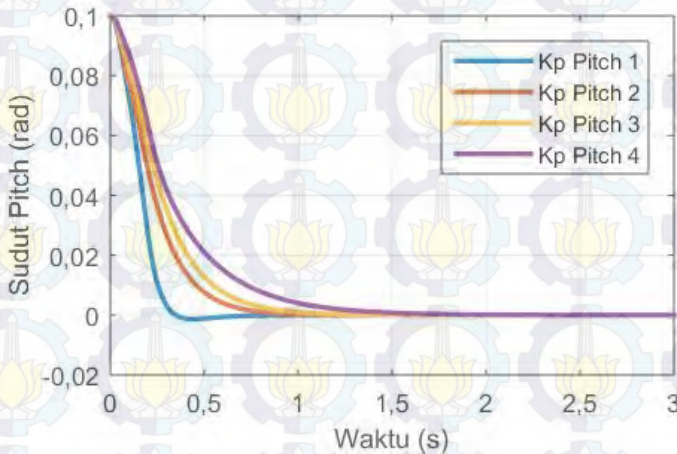
4.1.2 Simulasi Pengujian PID Pengendalian Sudut Pitch dengan Nilai Kp Berbeda

Pada pengujian ini ditetapkan nilai Ki *pitch* ditetapkan sebesar 0,01 dan juga nilai Kd *pitch* sebesar 1200. Penentuan nilai parameter PID ini dilakukan dengan cara *procedural tuning*. Nilai Kp yang diuji ada 4 yang seperti pada Tabel 4.3

Tabel 4. 3 Variasi Nilai Kp pada PID Pengendalian Sudut *Pitch*

No	<i>Pitch</i>	Nilai Kp
1	Kp <i>Pitch</i> 1	100
2	Kp <i>Pitch</i> 2	80
3	Kp <i>Pitch</i> 3	70
4	Kp <i>Pitch</i> 4	50

Hasil respon dari variasi nilai Kp ditunjukkan pada Gambar 4.2 dan spesifikasi dari respon ditunjukkan pada Tabel 4.4



Gambar 4. 2 Respon Variasi Nilai Kp pada Pengendalian PID Sudut *Pitch*

Tabel 4. 4 Karakteristik Respon Variasi Nilai Kp pada PID Pengendalian Sudut *Pitch*

Variasi Kp	<i>Time Constant</i> (s)	<i>Settling Time</i> (s)	<i>Rise Time</i> (s)
Kp <i>Pitch</i> 1	0,35	1,05	1,03
Kp <i>Pitch</i> 2	0,39	1,17	1,15
Kp <i>Pitch</i> 3	0,42	1,26	1,24
Kp <i>Pitch</i> 4	0,56	1,68	1,64
Ket: t_s ($\pm 5\%$), t_r (5%-95%)			

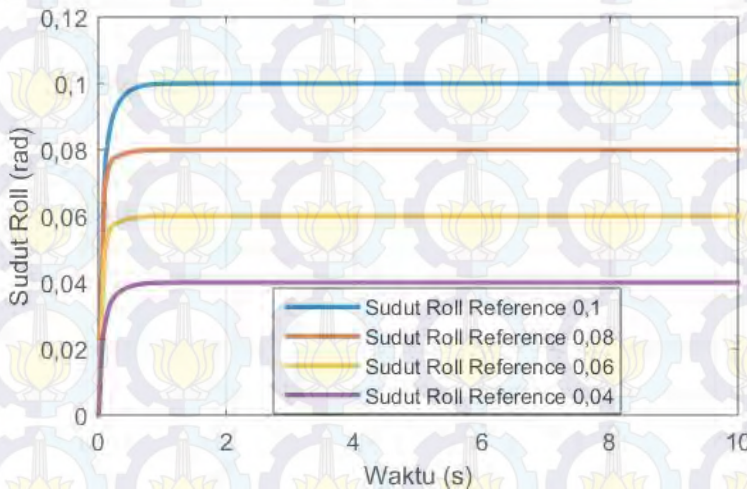
Dari respon variasi nilai Kp pada sudut *pitch* sesuai dengan Gambar 4.2 dan Tabel 4.4 terlihat bahwa nilai Kp *Pitch* 1 memiliki respon paling cepat menuju *steady state*. Sudut *pitch* yang dibutuhkan pada *plant quadcopter* ini diusahakan mempunyai respon rotasi yang cepat. Oleh karena itu nilai Kp yang dipilih adalah 100.

4.2 Simulasi Pengujian PID untuk Gerak Rotasi *Quadcopter*

Gerak rotasi merupakan gerakan *quadcopter* pada porosnya. Gerak rotasi terdiri dari 3 macam, yaitu gerak rotasi sudut *roll*, gerak rotasi sudut *pitch*, dan gerak rotasi sudut *yaw*. Pada penelitian Tugas Akhir ini hanya menggunakan gerak rotasi sudut *roll* dan *pitch* karena gerak sudut *yaw* dianggap *rigid*.

4.2.1 Simulasi Pengujian PID untuk Pengendalian Sudut *Roll*

Simulasi pada sudut *roll* dilakukan dengan memberi referensi sinyal *step* dengan nilai maksimal 0,1 atau bernilai $\pm 5,73^\circ$. Pada penelitian ini, nilai perubahan sudut dibatasi bernilai maksimum 0,1 radian dengan asumsi bahwa nilai tersebut dipilih karena sudut rotasi dari *quadcopter* harus mendekati nol untuk menjaga agar *quadcopter* tidak jatuh.

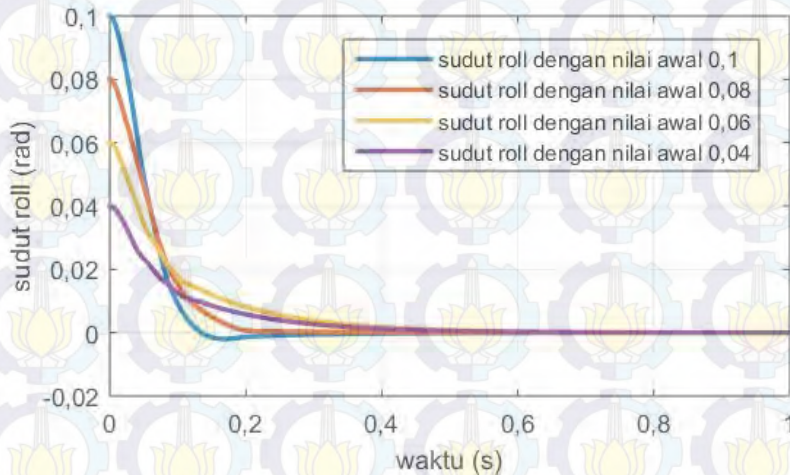


Gambar 4.3 Respon Simulasi Sudut *Roll* dengan Referensi Berbeda

Pada pengujian pertama diberi nilai referensi yang berbeda-beda. Jika respon yang diberikan mengikuti referensi tersebut, dapat dikatakan bahwa kontroler PID bekerja dengan baik.

Tujuan dari pengujian ini adalah untuk membuktikan bahwa sudut *roll* mampu mengikuti referensi yang diberikan, sehingga sesuai dengan Gambar 3.3 sudut *roll* akan mengikuti referensi dari sumbu Y. Pengujian

ini menggunakan sinyal step dengan variasi referensi 0,04 radian hingga 0,1 radian. Hasil dari pengujian terdapat pada Gambar 4.4



Gambar 4.4 Respon Simulasi Sudut Roll dengan Nilai Awal Berbeda

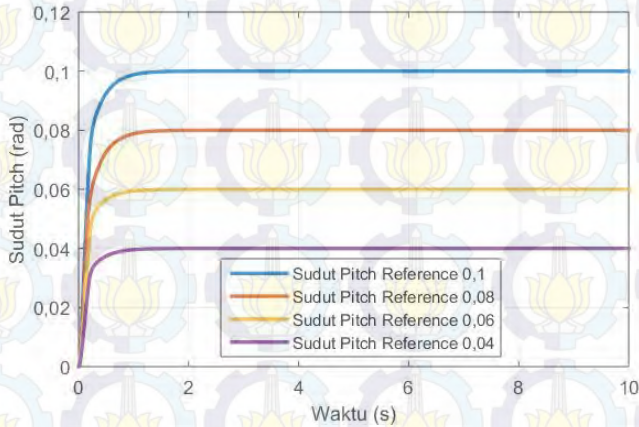
Selain itu juga dilakukan pengujian sudut roll dengan memberikan nilai awal yang berbeda-beda. Hasil pengujian terdapat pada Gambar 4.4. Tujuannya ialah untuk mengetahui apakah kontroler sudut roll tersebut dapat meregulasi ke titik awal. Jika kontroler tidak dapat meregulasi ke titik awal, maka *quadcopter* akan melakukan gerak translasi secara terus menerus dan menyebabkan *quadcopter* keluar dari lintasan (*trajectory*) yang telah ditentukan

Pada Gambar 4.3 dan Gambar 4.4 dapat diketahui bahwa pemberian referensi dan nilai awal yang berbeda tidak berpengaruh terhadap kestabilan sudut roll pada *quadcopter*. Dengan demikian bahwa performansi dari kontrol PID untuk pengendalian sudut roll yang telah dirancang dapat sesuai dengan performansi yang diinginkan untuk menstabilkan sudut roll dari *quadcopter*.

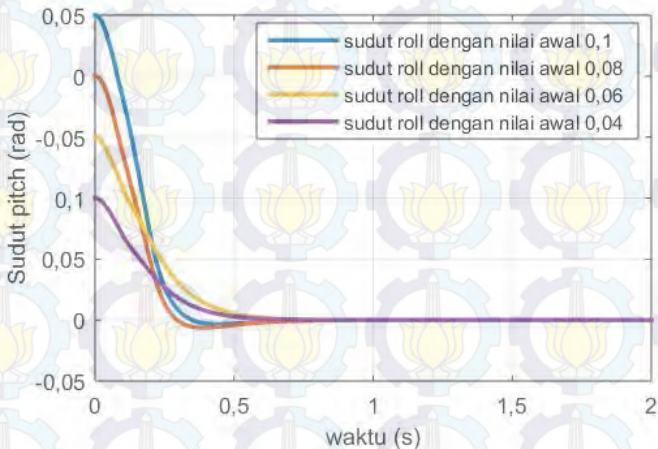
4.2.2 Simulasi Pengujian PID untuk Pengendalian Sudut Pitch

Pengujian sudut *pitch* dilakukan sama seperti pengujian sudut roll. Pengujian yang pertama dilakukan dengan memberikan nilai referensi yang berbeda. Pengujian sudut *pitch* terdapat pada Gambar 4.5 dan Gambar 4.6.

Pada Gambar 3.4 dijelaskan bahwa kontroler sudut *pitch* mendapat referensi dari kontroler gerak translasi sumbu X.



Gambar 4.5 Respon Simulasi Sudut *Pitch* dengan Referensi Berbeda



Gambar 4.6 Respon Simulasi Sudut *Pitch* dengan Nilai Awal Berbeda

Gambar 4.5 dan Gambar 4.6 dapat diketahui bahwa pemberian referensi dan nilai awal yang berbeda tidak berpengaruh terhadap

kestabilan sudut *pitch* pada *quadcopter*. Sehingga dapat dikatakan bahwa perancangan PID pengendalian sudut *pitch* telah sesuai dengan performansi yang diinginkan untuk menjaga kestabilan sudut *pitch* dari *quadcopter*

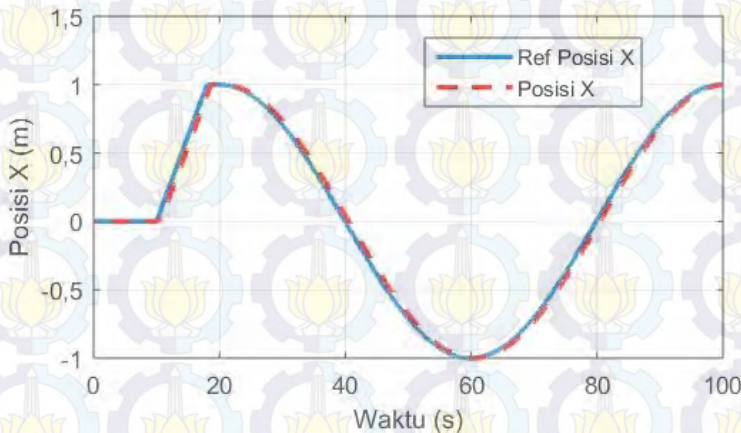
4.3 Simulasi Pengujian LQT dengan Nilai Q Berbeda pada Gerak *Cruise Quadcopter*

Pada penelitian Tugas Akhir ini digunakan metode LQT untuk pengendalian gerak translasi. Untuk mendapatkan nilai parameter Q dan R dari LQT digunakan metode *try and error*.

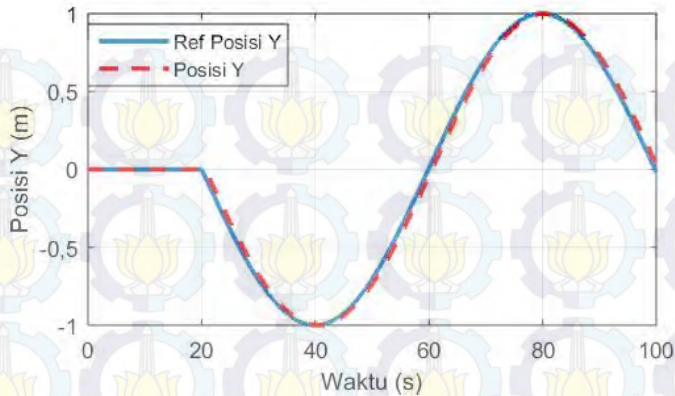
Pada pengujian ini dilakukan pemberian bobot Q yang berbeda dengan nilai bobot R 0,1 dengan tujuan mendapatkan nilai Q yang paling cocok. Pengujian ini menggunakan *signal builder* yang merepresentasikan pergerakan *quadcopter* dengan lintasan berbentuk lingkaran. Nilai bobot Q yang diujikan ditunjukkan pada Tabel 4.5

Tabel 4.5 Variasi Bobot Q Pada LQT Pengendalian Gerak *Cruise*

Variasi Q	Bobot Q
Q1	100
Q2	50
Q3	40



Gambar 4. 7 Respon Simulasi LQT Pergerakan *Quadcopter* pada Sumbu X dengan Menggunakan Bobot Q1



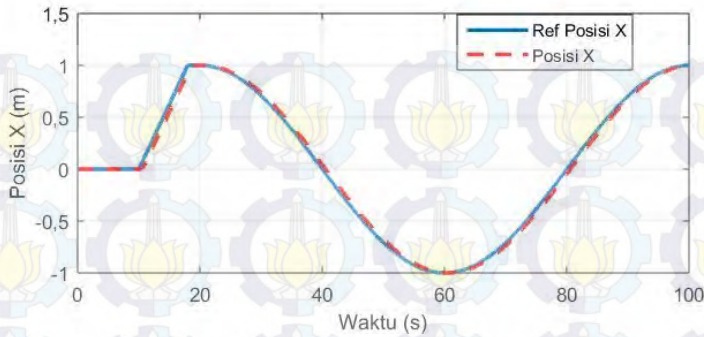
Gambar 4.8 Respon Simulasi LQT Pergerakan *Quadcopter* pada Sumbu Y dengan Menggunakan Bobot Q1

Dari hasil respon pada Gambar 4.7 dan 4.8 diatas dapat dihitung beberapa parameter hasil tuning bobot Q1 yang dicantumkan pada Tabel 4.6

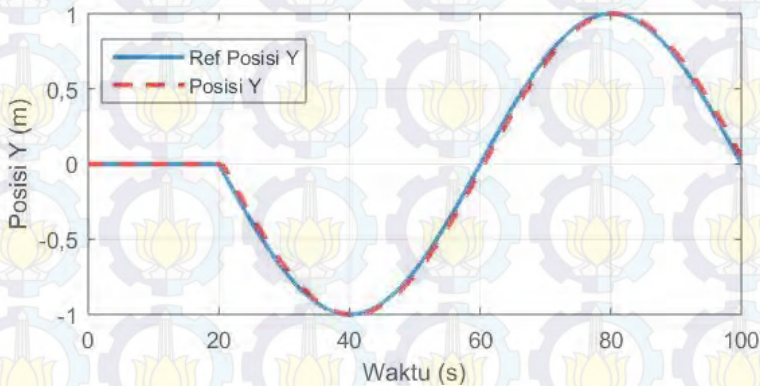
Tabel 4.6 Parameter Hasil *Tuning* dengan Bobot Q1

No	Hasil <i>Tuning</i> Q1	Nilai
1	RMSE _x	3,05%
2	RMSE _y	2,49%
3	ESS _x	0,1%
4	ESS _y	0,06%

Kemudian dilakukan simulasi LQT dengan nilai bobot Q2, dengan hasil seperti pada Gambar 4.9 untuk sumbu X, dan Gambar 4.10 untuk sumbu Y.



Gambar 4. 9 Respon Simulasi LQT Pergerakan *Quadcopter* pada Sumbu X dengan Menggunakan Bobot Q2



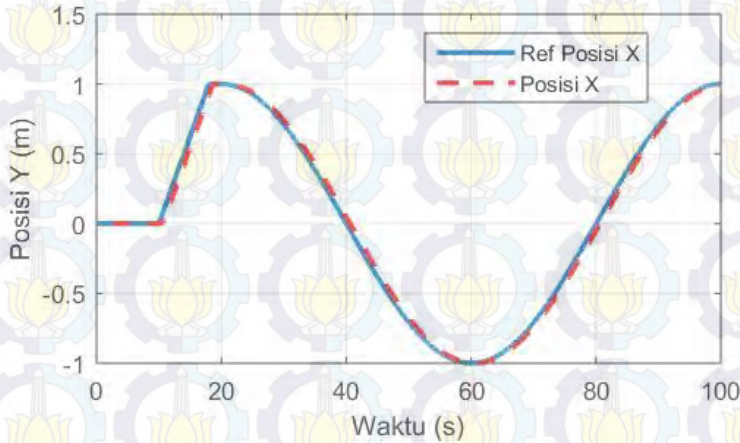
Gambar 4. 10 Respon Simulasi LQT Pergerakan *Quadcopter* pada Sumbu Y dengan Menggunakan Bobot Q2

Dari hasil respon diatas dapat dihitung beberapa parameter hasil tuning bobot Q2 yang dicantumkan pada Tabel 4.7

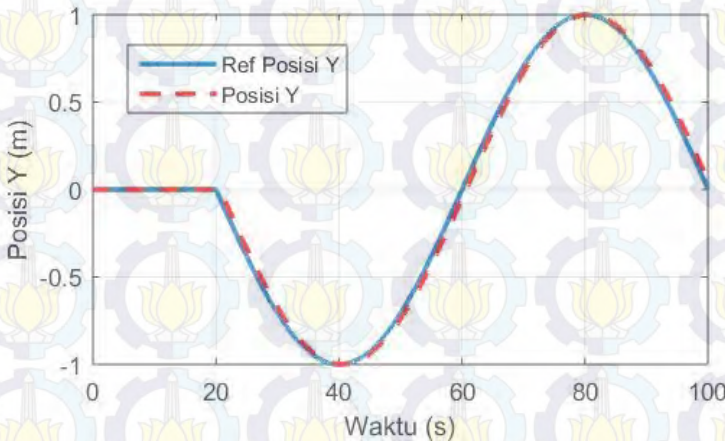
Tabel 4.7 Parameter Hasil *Tuning* Dengan Bobot Q2

No	Hasil <i>Tuning</i> Q2	Nilai
1	RMSE _x	3,63%
2	RMSE _y	2,95%
3	ESS _x	0,13%
4	ESS _y	0,08%

Kemudian dilakukan simulasi LQT dengan nilai bobot Q3, dengan hasil seperti pada Gambar 4.11 untuk sumbu X, dan Gambar 4.12 untuk sumbu Y.



Gambar 4.11 Respon Simulasi LQT Pergerakan *Quadcopter* pada Sumbu X dengan Menggunakan Bobot Q3



Gambar 4.12 Respon Simulasi LQT Pergerakan *Quadcopter* pada Sumbu Y dengan Menggunakan Bobot Q3

Dari hasil respon diatas dapat dihitung beberapa parameter hasil tuning bobot Q3 yang dicantumkan pada Tabel 4.8

Tabel 4.8 Parameter Hasil *Tuning* Dengan Bobot Q3

No	Hasil <i>Tuning</i> Q3	Nilai
1	RMSE _x	3,84%
2	RMSE _y	3,12%
3	ESS _x	0,15%
4	ESS _y	0.097%

Dari Tabel 4.6, Tabel 4.7, dan Tabel 4.8 dapat dibandingkan nilai nilai parameter yang didapatkan untuk membantu menentukan bobot Q yang digunakan. Berdasarkan ketiga tabel diatas bobot Q1 memiliki nilai *Error RMS* yang paling kecil dibanding yang lain yaitu 3,05% untuk sumbu X dan 2,49% untuk sumbu Y, sehingga bobot Q yang digunakan adalah Q1.

4.4 Simulasi Pengujian Gerak *Cruise Quadcopter* pada Lintasan Berbentuk Lingkaran dan Segi Empat

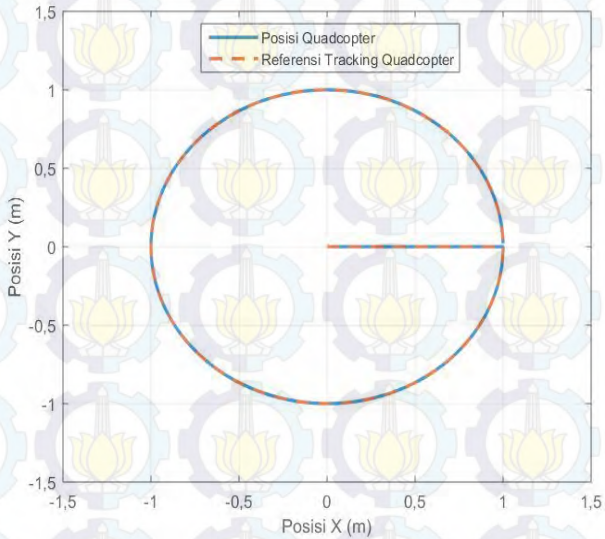
Penelitian Tugas Akhir ini membahas mengenai pengendalian *quadcopter* pada gerak *cruise* menggunakan kontroler LQT. Pengujian gerak *cruise* memiliki tujuan untuk memplot sumbu X dan sumbu Y dari pergerakan *quadcopter* sehingga membentuk lintasan yang diinginkan. Kontroler LQT harus bisa *men-tracking trajectory* yang telah diberikan.

Gerak *cruise* dapat dilaksanakan jika *quadcopter* dapat menjaga kestabilan di udara dan selanjutnya bergerak rotasi. Pengujian dilakukan dengan mensimulasikan sistem yang telah dirancang dengan menggunakan *signal builder*. Hal tersebut dikarenakan untuk *men-tracking* suatu *trajectory*, *quadcopter* memerlukan proses untuk menuju *trajectory* tersebut. Pada *signal builder* dapat dibuat sinyal sesuai dengan keinginan.

Prosedur dari pergerakan *quadcopter* adalah sebagai berikut:

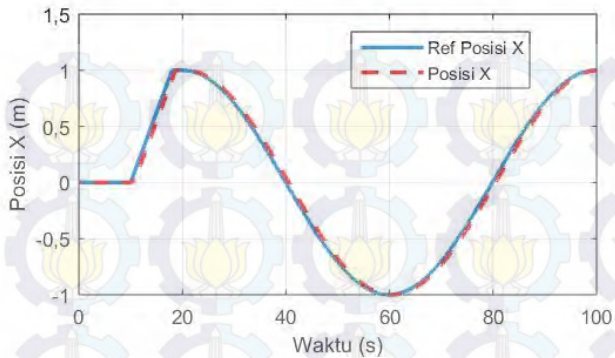
- Waktu simulasi yang digunakan adalah 100 detik
- Quadcopter* telah melakukan *take off* pada detik ke 10 dan terbang *hover* pada ketinggian 2 meter dengan kordinat (0,0).
- Pergerakan *quadcopter* dimulai dari kordinat (0,0) melewati lintasan lingkaran dan segi empat menuju kordinat (0,0) lagi setelah mengitari lintasan 1 kali dalam waktu 100 detik.
- Pengujian ini dilakukan dengan menghiraukan gangguan yang ada atau bersifat ideal. Hasil respon simulasi gerak *cruise*

quadcopter sesuai prosedur yang telah dibuat terdapat pada Gambar 4.13.

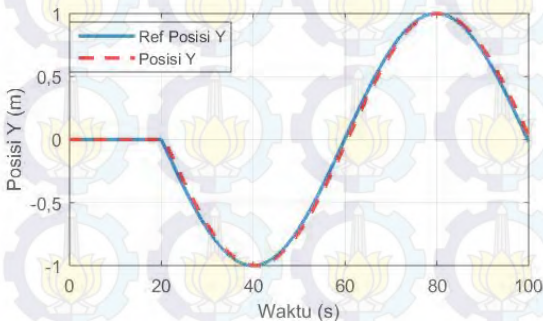


Gambar 4.13 Respon Simulasi Gerak *Cruise Quadcopter* pada Lintasan Berbentuk Lingkaran

Dari Gambar 4.13 dapat diketahui bahwa *quadcopter* mampu mengikuti lintasan lingkaran. Jika ditinjau pada kordinat $(0,0)$ respon dari *quadcopter* sudah sesuai dengan referensi. Garis biru yang ada dari koordinat $(0,0)$ hingga $(1,0)$ adalah posisi X ketika sumbu Z bergerak dari posisi) meter ke posisi 2 meter diatas. Untuk lebih jelas pergerakan *quadcopter* pada sumbu X yang Y dapat dilihat pada gambar 4.14 dan 4.15.



Gambar 4.14 Pergerakan *Quadcopter* pada Sumbu X dengan Lintasan Lingkaran



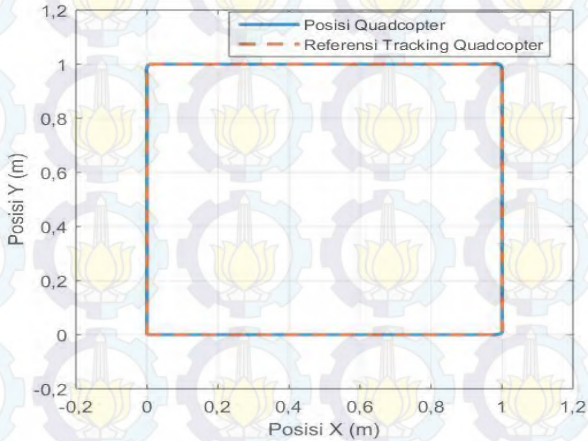
Gambar 4.15 Pergerakan *Quadcopter* pada Sumbu Y dengan Lintasan Lingkaran

Pada Gambar 4.14 dan Gambar 4.15 terlihat bahwa *quadcopter* mampu mengikuti pergerakan pada sumbu X maupun Y dengan simpangan yang sangat minim, terbukti dari parameter hasil pengujian seperti pada Tabel 4.9 yang menunjukkan nilai *error RMS* pada sumbu X hanya bernilai 3,05% serta *error RMS* pada sumbu Y bernilai 2,49%. Dari hasil hasil tersebut dapat dibilang bahwa pengujian dengan lintasan lingkaran ini berhasil.

Tabel 4.9 Parameter Pengujian *Quadcopter* Lintasan Lingkaran

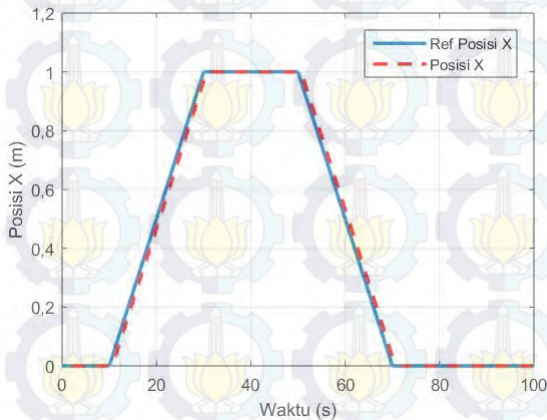
No	Parameter Pengujian	Nilai
1	RMSE _x	3,05%
2	RMSE _y	2,49%
3	ESS _x	0,1%
4	ESS _y	0,06%

Untuk selanjutnya *quadcopter* diuji dengan menggunakan lintasan berbentuk segi empat. Hasil yang diperoleh dari pengujian ini dapat dilihat dari Gambar 4.16.

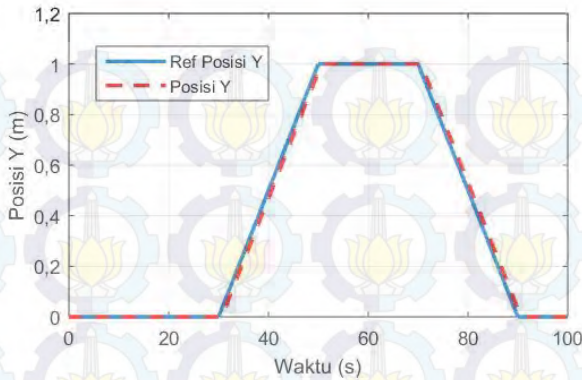


Gambar 4.16 Respon Simulasi Gerak *Cruise Quadcopter* pada Lintasan Berbentuk Segi Empat

Untuk lebih jelas pergerakan *quadcopter* pada sumbu X yang Y dapat dilihat pada Gambar 4.17 dan 4.18.



Gambar 4.17 Pergerakan *Quadcopter* pada Sumbu X dengan Lintasan Segi Empat



Gambar 4.18 Pergerakan *Quadcopter* pada Sumbu Y dengan Lintasan Segi Empat

Dari gambar diatas dapat dilihat bahwa *quadcopter* mampu mengikuti pergerakan pada sumbu X yang Y dengan lintasan berbentuk segi empat, terbukti dari parameter hasil pengujian seperti pada Tabel 4.10 yang menunjukkan nilai *error* RMS pada sumbu X dan sumbu Y hanya bernilai 1,58%. Dengan hasil yang diperoleh tersebut, dapat dibilang bahwa pengujian dengan lintasan segi empat berhasil.

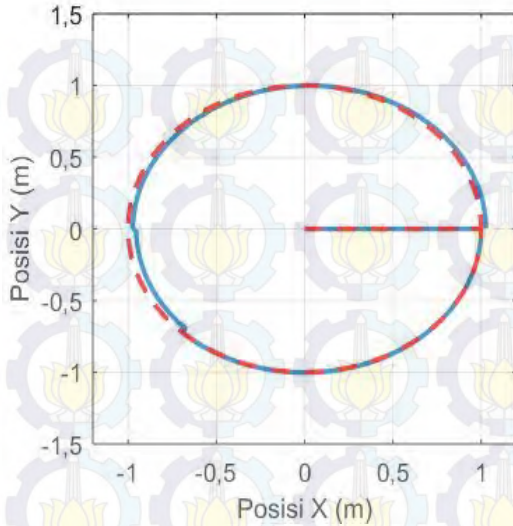
Tabel 4.10 Parameter Pengujian *Quadcopter* Lintasan Segi Empat

No	Parameter Pengujian	Nilai
1	RMSE _x	1,58%
2	RMSE _y	1,58%
3	ESS _x	0,02%
4	ESS _y	0,02%

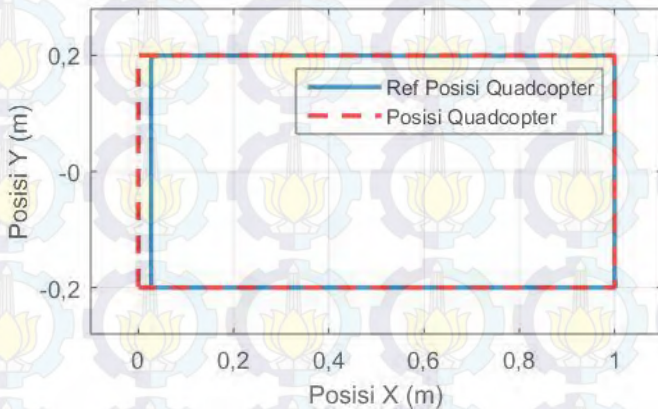
4.5 Simulasi Pengujian Gerak *Cruise Quadcopter* pada Lintasan Berbentuk Lingkaran dan Segi Empat dengan *Disturbance*

Untuk menguji kemampuan kontroler yang telah dirancang, pada pengujian simulasi ini ditambahkan *disturbance* pada sinyal kontrol. *Disturbance* yang digunakan merupakan sinyal *step* pada Matlab yang ditambahkan pada sinyal kontrol. Penambahan *disturbance* ini dapat direpresentasikan dengan angin yang mendorong *quadcopter*.

Hasil penambahan *disturbance* pada gerak *cruise quadcopter* terdapat pada Gambar 4. 19 dan Gambar 4.20



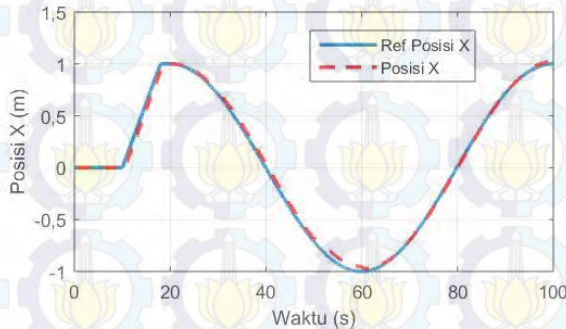
Gambar 4.19 Respon Simulasi Gerak *Cruise Quadcopter* pada Lintasan Berbentuk Lingkaran dengan *Disturbance*



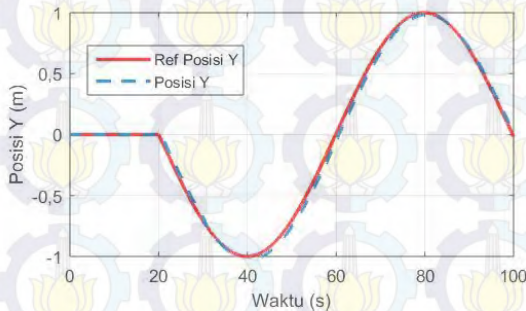
Gambar 4.20 Respon Simulasi Gerak *Cruise Quadcopter* pada Lintasan Berbentuk Segi Empat dengan *Disturbance*

Pada Gambar 4.19 terlihat bahwa penambahan *disturbance* tidak membuat *quadcopter* kehilangan keseimbangan. *Quadcopter* terlihat kembali menuju referensi meskipun diberi gangguan dengan masukan *disturbance* sebesar 50% dari gaya angkat minimal *quadcopter* pada U_1 . Pada Gambar 4.20,

terlihat pula dengan penambahan *disturbance* tidak mengakibatkan *quadcopter* kehilangan keseimbangan. *Quadcopter* terlihat kembali menuju *track* lagi. *Disturbance* tersebut mengakibatkan adanya *error* pada pergerakan *quadcopter* karena kontroler PID mendapatkan referensi secara terus menerus dari *disturbance*. Sehingga mengakibatkan terjadi perubahan sudut *quadcopter* yang mengakibatkan adanya *error* dalam pergerakan. Namun kontroler yang telah dirancang masih dapat mengendalikan *quadcopter* untuk kembali mengikuti referensi berbentuk lingkaran dan segi empat dengan simpangan yang kecil. Untuk lebih jelas pergerakan *quadcopter* pada sumbu X yang Y dapat dilihat pada Gambar 4.21 dan 4.22 untuk lintasan lingkaran, serta Gambar 4.23 dan Gambar 4.24 untuk lintasan segi empat.



Gambar 4.21 Pergerakan *Quadcopter* pada Sumbu X pada Lintasan Lingkaran dengan *Disturbance*

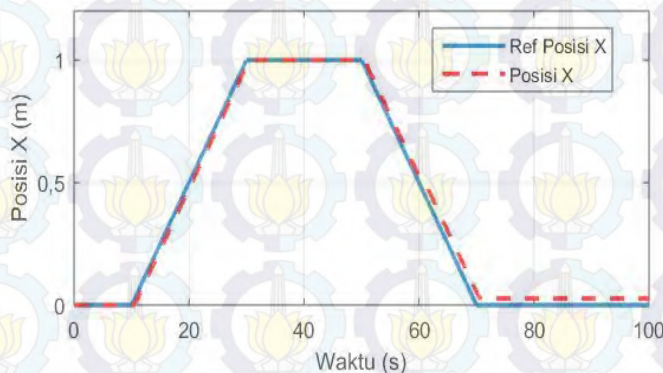


Gambar 4.22 Pergerakan *Quadcopter* pada Sumbu Y pada Lintasan Lingkaran dengan *Disturbance*

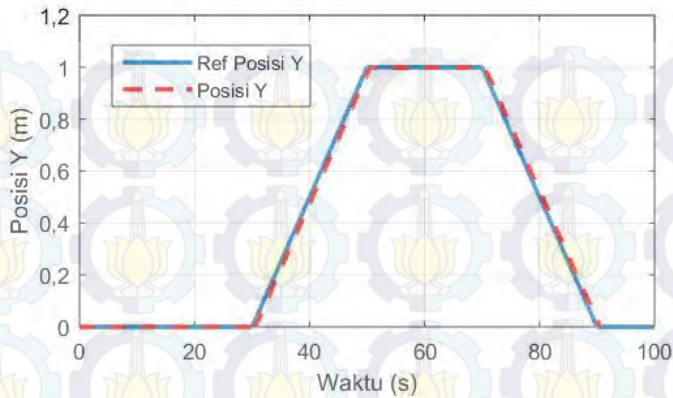
Pada Gambar 4.21 dan 4.22 dapat dilihat bahwa *quadcopter* mengalami pergeseran akibat dari adanya *disturbance*. Namun gangguan itu tidak mengakibatkan *quadcopter* kehilangan keseimbangan. *Quadcopter* terlihat kembali menuju lintasan, serta tetap mampu menyelesaikan *tracking* dengan mengikuti referensi yang diberikan dengan hanya memiliki *error RMS* sumbu X sebesar 3,18% dan *error RMS* pada sumbu Y sebesar 2,53% seperti yang ditunjukkan pada tabel 4.11.

Tabel 4.11 Parameter Pengujian *Quadcopter* Lintasan Lingkaran dengan *Disturbance*

No	Parameter Pengujian	Nilai
1	RMSE _x	3,18%
2	RMSE _y	2,53%
3	ESS _x	0,1%
4	ESS _y	0,06%



Gambar 4.23 Pergerakan *Quadcopter* pada Sumbu X dengan Lintasan Segi Empat dengan *Disturbance*



Gambar 4.24 Pergerakan *Quadcopter* pada Sumbu Y dengan Lintasan Segi Empat dengan *Disturbance*

Pada Gambar 4.23 dan 4.24 dapat dilihat bahwa *quadcopter* mengalami sedikit pergeseran akibat dari adanya *disturbance*. Namun gangguan itu tidak mengakibatkan *quadcopter* kehilangan keseimbangan. *Quadcopter* terlihat kembali menuju lintasan, serta *quadcopter* tetap mampu menyelesaikan *tracking* dengan mengikuti referensi yang diberikan dengan hanya memiliki *error* RMS sumbu X sebesar 2,6% serta *Error* RMS sumbu Y sebesar 1,61% seperti yang ditunjukkan pada Tabel 4.12.

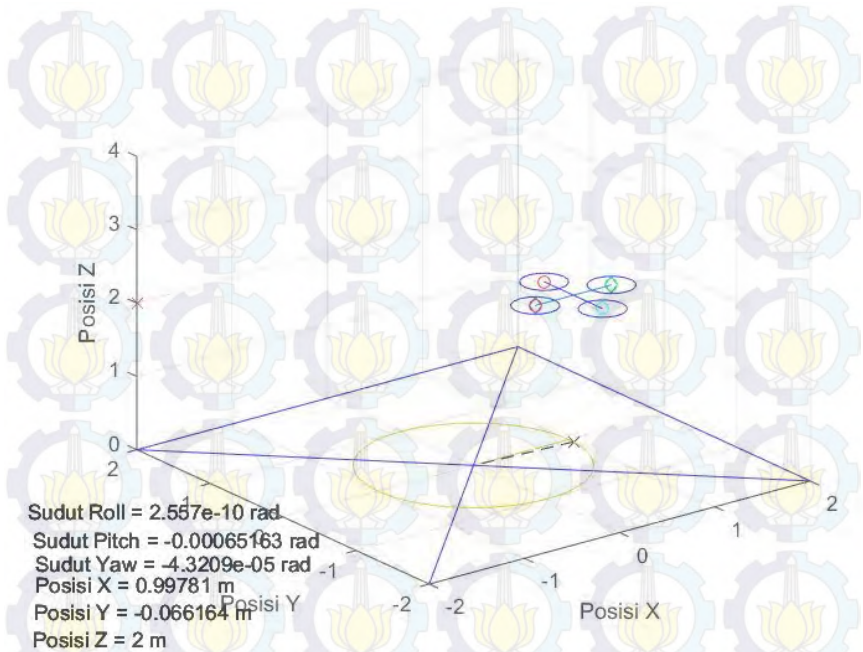
Tabel 4.12 Parameter Pengujian *Quadcopter* Lintasan Segi Empat dengan *Disturbance*

No	Parameter Pengujian	Nilai
1	RMSE _x	2,6%
2	RMSE _y	1,61%
3	ESS _x	0,07%
4	ESS _y	0,03%

4.6 Simulasi 3D Gerak *Cruise Quadcopter* pada Lintasan Berbentuk Lingkaran dan Segi Empat

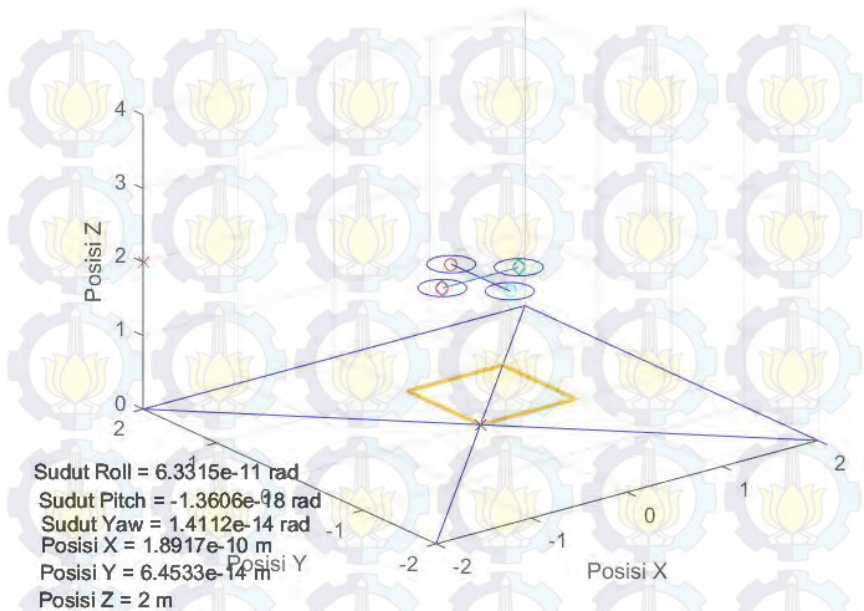
Simulasi ini digunakan simulasi 3D untuk mengetahui pergerakan *quadcopter* yang sesungguhnya. Simulasi 3D ini menggunakan *trajectory* berbentuk lingkaran dan segi empat. Pada simulasi 3D dibatasi sumbu X, Y, Z bernilai masing-masing 2 meter, 2 meter, dan 4 meter untuk

mempermudah dalam pembuatan dan ditunjukkan pada Gambar 4.25 dan Gambar 4.26.



Gambar 4. 25 Simulasi *Tracking 3D Quadcopter* Lintasan Lingkaran Beserta Lintasannya

Pada Gambar 4.25 terlihat *quadcopter* ketika berada pada posisi koordinat (1,0,2), posisi dimana ketika *quadcopter* telah melakukan gerak *cruise*. Dan pada titik yang sama itu pula *quadcopter* berhenti simulasi. *Start* simulasi *quadcopter* dimulai pada posisi koordinat (0,0,0).



Gambar 4.26 Simulasi 3D *Tracking Quadcopter* Lintasan Segi Empat Beserta Lintasannya

Pada Gambar 4.26 terlihat *quadcopter* ketika berada pada posisi koordinat (0,0,2), posisi dimana ketika *quadcopter* telah melakukan gerak *hover*. Dan pada titik yang sama itu pula *quadcopter* berhenti simulasi. *Start* simulasi *quadcopter* dimulai pada posisi koordinat (0,0,0).



--halaman ini sengaja dikosongkan--

BAB V PENUTUP

Hasil dari perancangan dan penelitian Tugas Akhir dirangkum dan dirumuskan kesimpulan. Kesimpulan ini menerangkan hasil dari pengujian dan simulasi yang telah dilaksanakan.

Selama proses perancangan dan penelitian, terdapat banyak kendala yang dihadapi. Kendala tersebut telah penulis rangkum dan dirumuskan dalam bentuk saran untuk penyempurnaan dan penelitian lebih lanjut

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisis, dapat diperoleh kesimpulan sebagai berikut:

- a) Untuk dapat membuat *quadcopter* mampu bergerak dengan gerak rotasi dan translasi yang selaras, kecepatan respon gerak rotasi harus lebih cepat daripada respon gerak translasi.
- b) Kontroler PID dapat digunakan untuk mengendalikan gerak rotasi dengan baik, karena mampu menghasilkan respon yang lebih cepat daripada respon gerak translasi.
- c) Metode *Linear Quadratic Tracking* dengan *tuning* menggunakan *try and error* dapat digunakan untuk mengendalikan gerak *cruise quadcopter* pada lintasan berbentuk lingkaran dan segi empat
- d) Hasil *tuning try and error* kontroler LQT didapatkan nilai bobot Q dan R masing-masing bernilai 100 dan 0,1. Dengan hasil nilai RMSE_x 3,05% dan RMSE_y 2,49% untuk lintasan lingkaran, sedangkan untuk lintasan segi empat mempunyai RMSE_x dan RMSE_y dengan nilai yang sama yaitu 1,58%.
- e) Kontroler yang telah dirancang belum dapat diimplementasikan karena Ardupilot 2.6 tidak dapat diubah *script*-nya dan belum dapat membentuk komunikasi antara MATLAB dengan kontroler.



5.2 Saran

Dari hasil penelitian yang dilakukan, untuk pengembangan berikutnya, disarankan beberapa hal berikut ini:

- a) Pemodelan dan pemahaman tentang *quadcopter* akan membantu perancangan sistem yang lebih baik dan mendapatkan model matematika yang lebih akurat.
- b) Penggunaan program PID dan LQT masih bisa didapatkan nilai yang lebih optimal dengan mencari parameter tiap kontroler dengan *tuning* tertentu.
- c) Komunikasi antara MATLAB dengan kontroler diselesaikan agar dapat memudahkan ketika akan diimplementasikan.

DAFTAR PUSTAKA

- [1] Tommaso Bresciani, "Modelling, Identification and Control of a Quadrotor Helicopter". *Thesis*, Department of Automatic Control Lund University, 2008
- [2] Spong, M.W, Vidyasagar, M. "*Robot Dynamic and Control*", Prentice Hall, New Jersey, 2004
- [3] Akbar, Farid Choirul,"Tuning Parameter Linear Quadratic Tracking Menggunakan Algoritma Genetika Untuk Pengendalian Gerak Lateral Quadcopter". *Tugas Akhir Teknik Elektro ITS*, Surabaya: Institut Teknologi Sepuluh Nopember, 2016
- [4] Singhal, Rahul, P.B., Sujit, "3D Trajectory Tracking For a Quadcopter using MPC on a 3D Terrain", *IEEE Transaction on Aerospace and Electrical Systems*, 2015
- [5] Yuliansyah, Rendy,,"Desain dan Implementasi Kontroler PID untuk Glide-Scope Tracking pada Fixed-Wing UAV (Unmanned Aerial Vehicle)". *Tugas Akhir Teknik Elektro ITS*, Surabaya: Institut Teknologi Sepuluh Nopember, 2011.
- [6] Naidu, Subbaram D, "*Optimal Control Systems*", CRC Press Idaho, Ch.3-4, 2002
- [7] H.K. Khalil and JW Grizzle, "*Nonlinear systems*", Prentice Hall, New Jersey, 2002
- [8] Skogestad, Sigurd., "Probably The Best Simple PID Tuning Rules in The World". *Journal of Process Control*, Norwegia: Department of Chemical Engineering, Norwegian University of Science and Technology, 2001



LAMPIRAN A

A1. Pengukuran Kecepatan Motor [8]

Pulsa	Motor 1	Motor 2	Motor 3	Motor 4
75	114.505	111.888	109.376	113.668
80	152.29	152.604	148.836	154.069
85	187.563	192.900	187.772	193.633
90	219.276	225.242	221.684	229.429
95	252.77	258.108	256.747	263.76
100	281.03	290.973	287.31	295.578
105	313.267	325.722	322.373	330.223
110	342.26	355.552	351.156	360.262
115	369.892	384.964	380.986	390.93
120	399.198	416.468	412.386	422.958
125	427.04	447.031	440.646	452.16
130	453.625	471.732	466.29	475.396
135	479.164	500.411	496.957	508.575
140	502.295	503.97	519.042	533.172
145	525.531	523.752	543.848	558.292
150	546.569	548.872	566.037	582.784
155	570.538	572.212	574.410	606.334
160	590.529	595.344	592.832	630.721
165	611.567	615.126	609.997	654.271
170	633.024	629.465	627.790	656.050
175	651.236	653.434	650.817	661.702
180	673.111	670.076	672.378	673.844
185	676.356	672.169	675.937	676.565

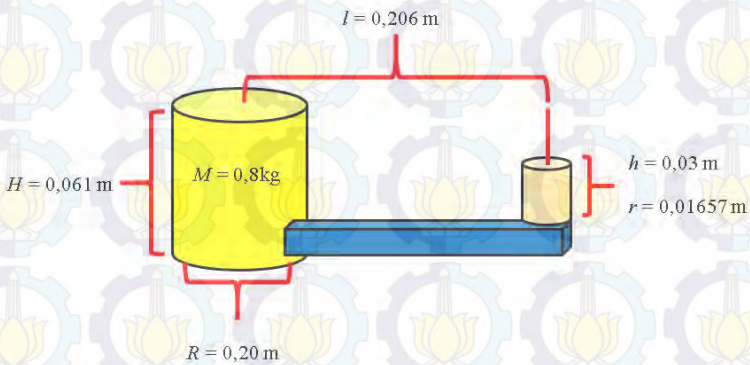
Pulsa	Motor 1	Motor 2	Motor 3	Motor 4
190	673.006	672.483	673.948	675.832
195	672.902	674.053	673.00	675.1
200	673.006	674.262	672.797	675.623

A2 Pengukuran Gaya Angkat Motor [8]

Pulsa	Gaya Angkat 1 (kg)	Gaya Angkat 2 (kg)	Gaya Angkat 3 (kg)	Gaya Angkat 4 (kg)
75	0.02	0.011	0.017	0.014
80	0.036	0.031	0.033	0.039
85	0.058	0.055	0.055	0.061
90	0.078	0.08	0.077	0.084
95	0.104	0.107	0.102	0.115
100	0.131	0.138	0.134	0.145
105	0.165	0.177	0.17	0.184
110	0.198	0.207	0.209	0.221
115	0.227	0.252	0.243	0.264
120	0.27	0.297	0.285	0.306
125	0.31	0.34	0.328	0.357
130	0.352	0.39	0.374	0.402
135	0.396	0.432	0.426	0.452
140	0.438	0.446	0.478	0.498
145	0.488	0.475	0.522	0.553
150	0.533	0.54	0.578	0.611
155	0.582	0.605	0.613	0.664
160	0.63	0.626	0.619	0.718
165	0.672	0.654	0.646	0.771

Pulsa	Gaya Angkat 1 (kg)	Gaya Angkat 2 (kg)	Gaya Angkat 3 (kg)	Gaya Angkat 4 (kg)
170	0.723	0.715	0.704	0.783
175	0.772	0.784	0.762	0.791
180	0.834	0.832	0.826	0.825
185	0.842	0.846	0.84	0.847
190	0.833	0.833	0.836	0.841
195	0.832	0.834	0.834	0.845
200	0.834	0.836	0.837	0.845

A3. Identifikasi Fisik *Quadcopter* [8]



Identifikasi fisik konstanta

$$I_{z,frame1,4} = \frac{m_l}{12}(p^2 + t^2)$$

$$I_{z,frame2,3} = \frac{m_l}{12}(p^2 + t^2)$$

$$I_{z,m1,2,3,4} = \frac{m_m}{2} r^2 + (p_1)^2 \cdot M$$

$$I_{z.batt} = \frac{m_b}{12}(p^2 + l^2)$$

$$I_{z.1} = \frac{m_{mika}}{12}(p^2 + l^2)$$

$$I_{z.2} = \frac{m_{mika}}{12}(2.l^2)$$

$$I_{z.1,2,3,4} = \frac{m_p}{2} r^2 + (p_1)^2 \cdot M_p$$

$$I_{x.frame1,2,3} = \frac{m_l}{12}(l^2 + t^2)$$

$$I_{x.frame4,5,6} = \frac{m_l}{12}(p^2 + l^2)$$

$$I_{z.m2,4} = m_m \left(\frac{r^2}{4} + \frac{h}{12} \right) + [(0.5h)^2 + (p_1 - p_2)^2] m$$

$$I_{z.m1,3} = m_m \left(\frac{r^2}{4} + \frac{h}{12} \right) + (0.5h)^2 m$$

$$I_{x.batt} = \frac{m_b}{12}(l^2 + t^2) + (0.5t)^2$$

$$I_{y.batt} = \frac{m_b}{12}(p^2 + t^2) + (0.5t)^2$$

$$I_{x.mika1} = \frac{m_{mika}}{12}(p^2 + t^2) + (t^2) m_{mika}$$

$$I_{x.mika2,3} = \frac{m_{mika}}{12}(l^2 + t^2) + (t^2) m_{mika}$$

$$I_{y.mika1} = \frac{m_{mika}}{12}(l^2 + t^2) + (t^2) m_{mika}$$

$$I_{x.mika2,3} = \frac{m_{mika}}{12}(l^2 + t^2) + (t^2 + l^2) m_{mika}$$

$$I_{x.1,2} = m_p \left(\frac{p_2^2}{6} + \frac{t^2}{12} + p_1^2 + (0.5t^2 + h)^2 \right)$$

$$I_{x,3,4} = m_p \left(\frac{p_2^2}{6} + \frac{t^2}{12} \right)$$

Perhitungan I_{xx} , I_{yy} , I_{zz} dan J_{TP}

$$I_{xx} = \frac{mr^2}{4} + \frac{mh^2}{6} + 2mr^2 + \frac{MR^2}{4} + \frac{MH^2}{12}$$

$$I_{xx} = 1,68 \times 10^{-3}$$

$$I_{yy} = \frac{mr^2}{4} + \frac{mh^2}{6} + 2mr^2 + \frac{MR^2}{4} + \frac{MH^2}{12}$$

$$I_{yy} = 1,68 \times 10^{-3}$$

$$I_{zz} = \frac{MR^2}{2} + 4mr^2$$

$$I_{zz} = 1,25 \times 10^{-3}$$

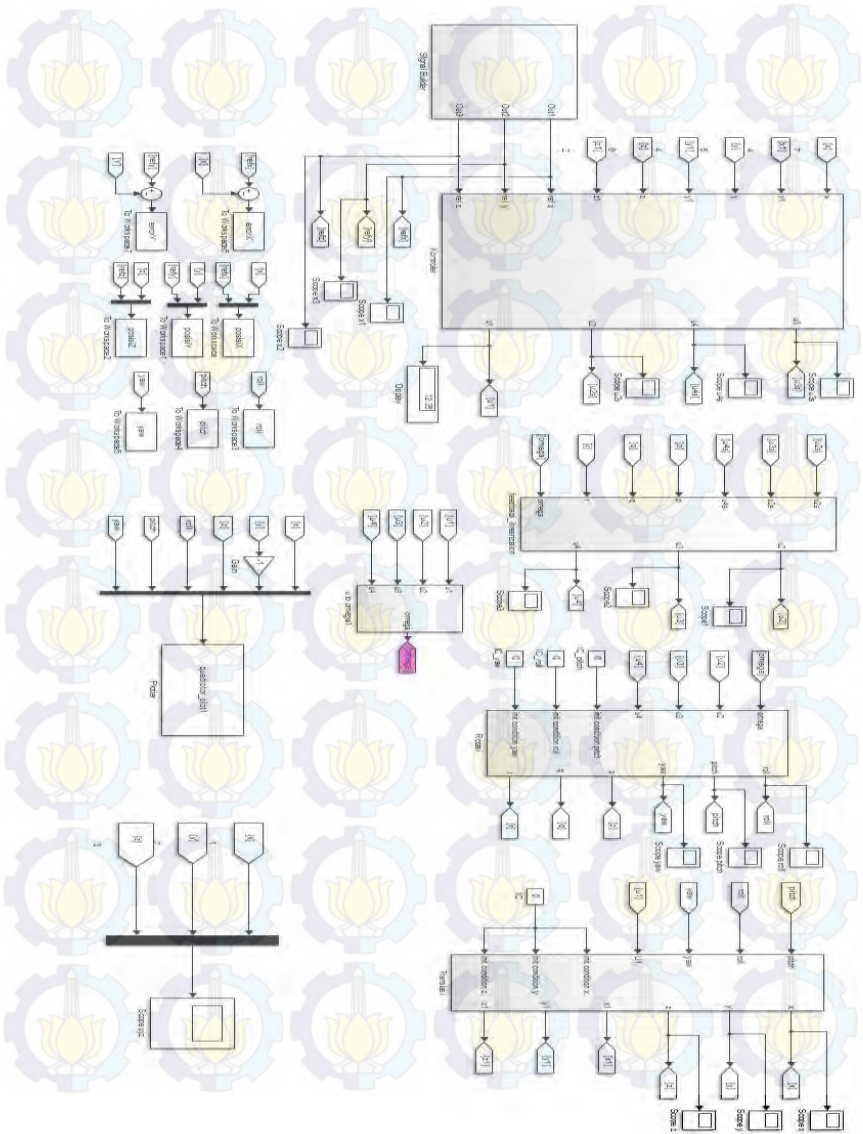
$$J_{TP} = \frac{4 \times (mr^2)}{2}$$

$$J_{TP} = 0,21175 \times 10^{-4}$$

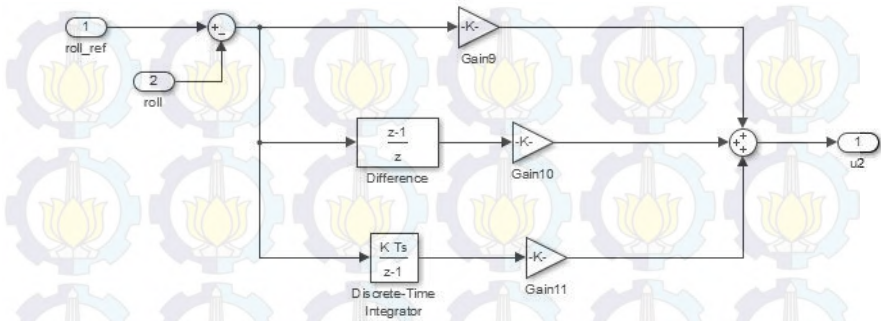


LAMPIRAN B

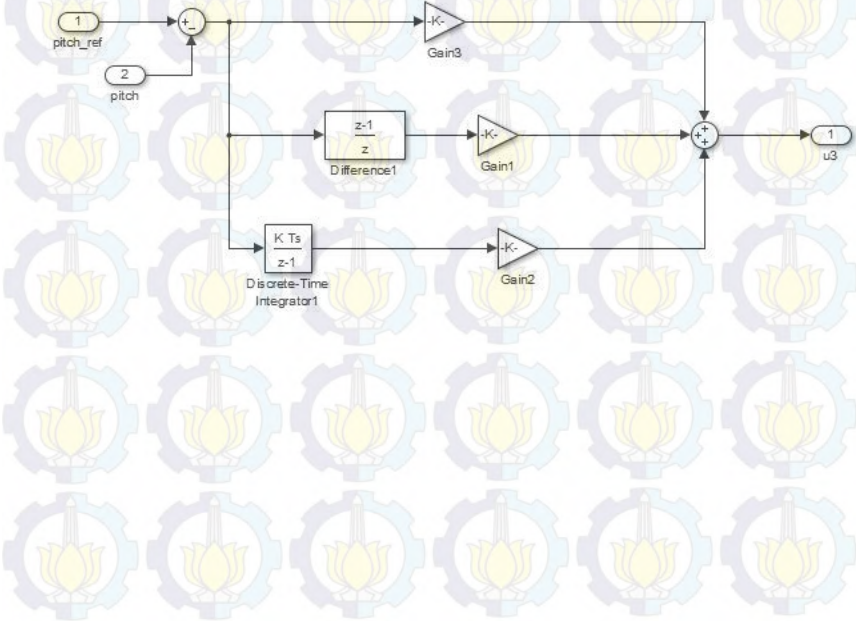
B1. Simulink Keseluruhan Sistem



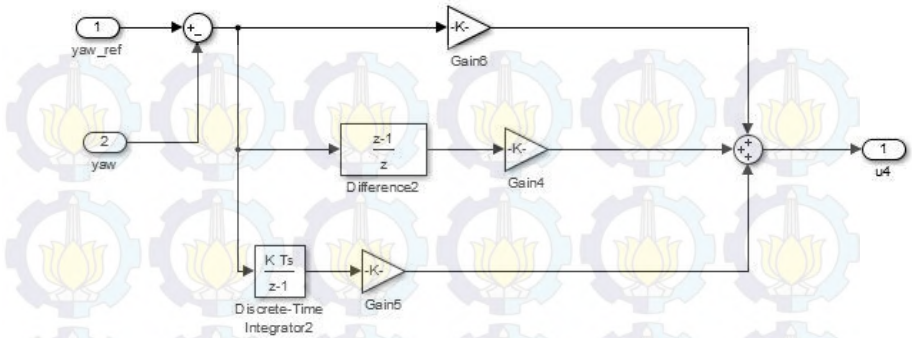
B2 Simulink Pengendalian Sudut Roll



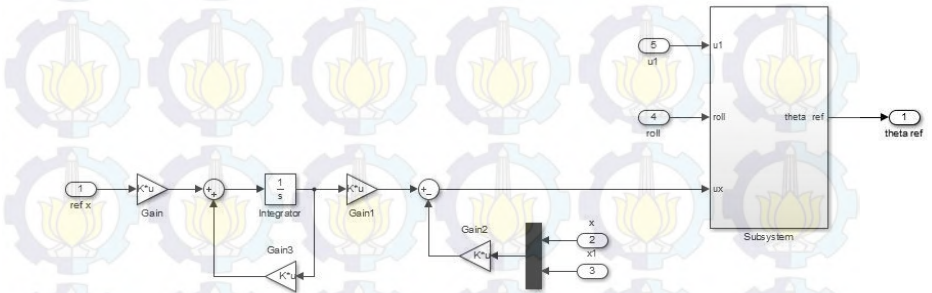
B3. Simulink Pengendalian Sudut Pitch



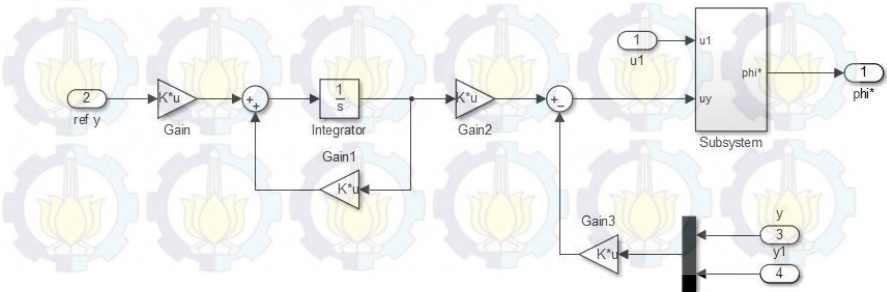
B4. Simulink Pengendalian Sudut Yaw



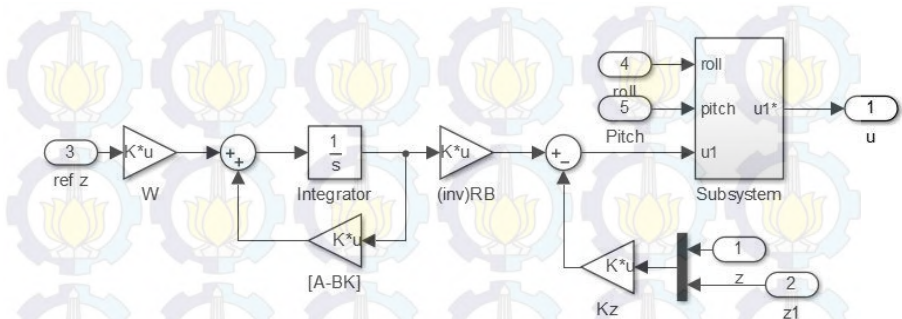
B5. Simulink Pengendalian Posisi X



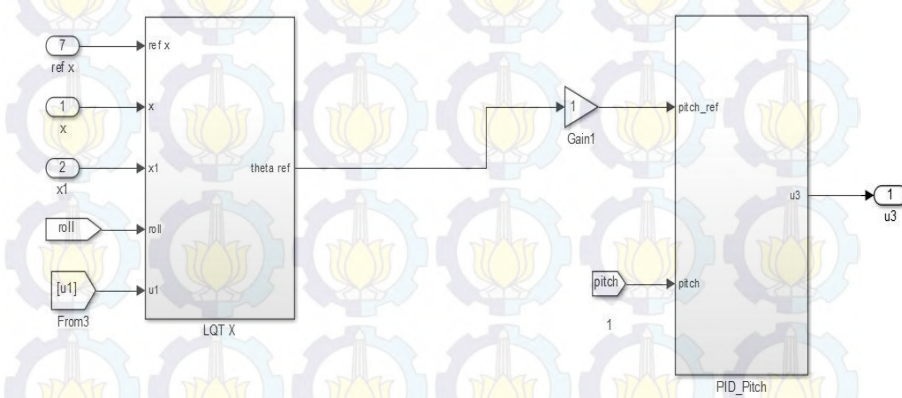
B6. Simulink Pengendalian Posisi Y



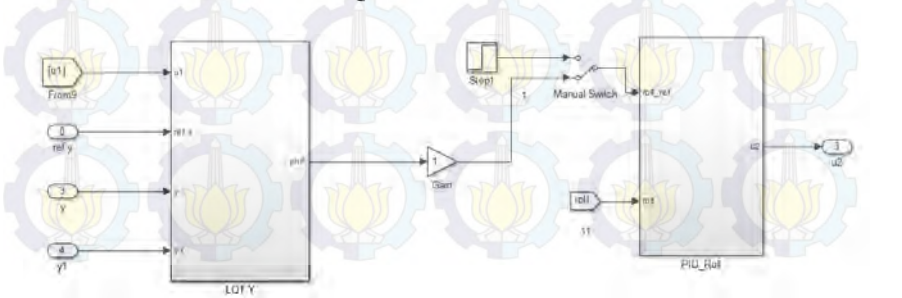
B7. Simulink Pengendalian Posisi Z



B8. Simulink Cascade LQT PID Sumbu X



B9. Simulink Cascade LQT PID Sumbu Y



LAMPIRAN C

C1. Program PID LQT pada MATLAB

```
%parameter linearisasi rotasi plant!
```

```
%parameter pdot!
```

```
a11=-0.5495;
```

```
b11=0.0017;
```

```
c11=0.2052;
```

```
%parameter qdot!
```

```
a21=0.1675;
```

```
b21=-0.0094;
```

```
c21=2.955;
```

```
%parameter rdot
```

```
a31=-2.0257;
```

```
b31=0.0594;
```

```
%parameter quadcopter!
```

```
m1=1.26;
```

```
%Parameter PID
```

```
Kp_pitch=100;
```

```
Kp_roll=800;
```

```
Kp_yaw=50;
```

```
Ki_pitch=0.01;
```

```
Ki_roll=0.01;
```

```
Ki_yaw=0.0001;
```

```
Kd_pitch=1200;
```

```
Kd_roll=7000;
```

```
Kd_yaw=950;
```

```
%LQT!
```

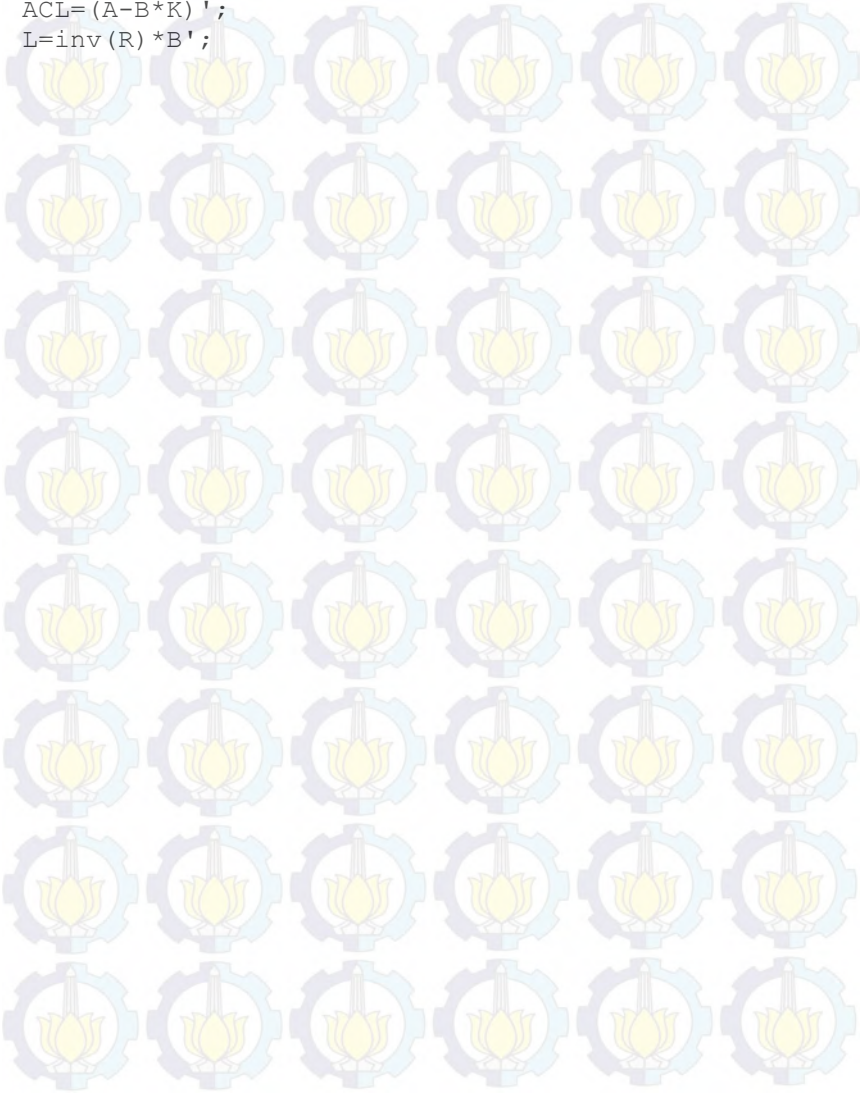
```
A=[0 1;0 0];
```

```
B=[0 1]';
```

```
C=[1 0];
```

```
Q=100;
```

```
R=0.1;  
[S,o,m,n]=care(A,B,C'*Q*C,R);  
K=inv(R)*B'*S;  
ACL=(A-B*K)';  
L=inv(R)*B';
```



RIWAYAT PENULIS



Mohammad Faizal Shultoni, biasa dipanggil Faizal oleh teman-temannya lahir di Blera, 10 Juni 1994. Faizal merupakan anak kedua dari pasangan M. Syaiful Anam dan Suntin Malika. Lulus dari SDN 03 Cepu pada tahun 2006, kemudian melanjutkan studi ke jenjang lebih lanjut di SMPN 3 Cepu dan lulus pada tahun 2009. Kemudian melanjutkan ke SMAN 1 Cepu dan lulus pada tahun 2012. Setelah menembuh studi pada tingkat SMA, penulis melanjutkan ketingkat lebih lanjut, yaitu di Institut Teknologi Sepuluh Nopember Surabaya jurusan Teknik Elektro pada tahun 2012 dan memutuskan untuk mengambil bidang studi Sistem Pengaturan.



--halaman ini sengaja dikosongkan--